

BAB IV

ANALISIS DAN PERANCANGAN SISTEM

Seperti yang telah dijelaskan pada gambar 3.1, proses pengembangan sistem dalam penelitian ini menggunakan model proses *Waterfall*. Tahapan-tahapan pengembangan sistem yang dilakukan adalah :

1. Definisi dan Analisis Kebutuhan (*Requirements Analysis and Definition*),
2. Desain/ Perancangan Sistem (*System and Software Design*),
3. Implementasi dan Pengujian Unit (*Implementation and Unit Testing*),
4. Integrasi dan Pengujian Sistem (*Integration and System Testing*), dan
5. Operasi dan Pemeliharaan (*Operation and Maintenance*).

Pendekatan pengembangan sistem yang digunakan untuk implementasi algoritma adalah pendekatan terstruktur. Pendekatan terstruktur dilakukan untuk mendapatkan sistem dengan struktur pendefinisian yang baik dan jelas. Selain itu, pendekatan ini juga untuk memecahkan permasalahan yang kompleks, pemeliharaan yang mudah, dan mempunyai dokumentasi yang baik. Berikut merupakan penjelasan dari tahapan analisis dan perancangan sistem.

4.1 Definisi dan Analisis Kebutuhan

4.1.1 Analisis Kebutuhan Perangkat

Perangkat-perangkat yang digunakan dalam penelitian ini adalah :

1. Perangkat Keras

Dalam pengembangan sistem, salah satu pendukung keberhasilan sistem adalah faktor pemakaian perangkat keras yang memadai agar algoritma dapat berjalan secara minimum. Perangkat keras yang digunakan dalam penelitian ini adalah satu unit laptop Dell Vostro1014 dengan spesifikasi :

- a. Processor : Intel® Core™ Duo CPU T5870 @2.00GHz
- b. Memori : 952 MB of RAM
- c. Monitor 14", Keyboard and Mouse

2. Perangkat Lunak

Perangkat Lunak yang dibutuhkan untuk aplikasi algoritma adalah :

- a. Sistem Operasi Windows sebagai sistem utama yang dapat menjalankan perangkat lunak lainnya. Dalam hal ini, penulis menggunakan Windows 7
- b. MySQL sebagai penyimpanan database.
- c. Bahasa Pemrograman Visual Basic 6.0 sebagai bahasa pemrograman pengembangan sistem

4.1.2 Analisis Kebutuhan Sistem

Kebutuhan sistem dalam penelitian ini adalah :

1. Kebutuhan Masukan (*Input*)

Kebutuhan masukan (*input*) merupakan data masukan yang berasal dari pengguna sistem, data tersebut yaitu

a. Data simpul, yang meliputi jumlah simpul dan koordinat simpul

b. Parameter Algoritma Genetika, yaitu

- Ukuran Populasi
- Peluang *Crossover* ($0 < P_c < 1$)
- Peluang Mutasi ($0 < P_m < 1$)
- Peluang kelestarian ($0 < K_b < 1$)

2. Kebutuhan Proses

Proses-proses yang terjadi dalam penelitian ini digambarkan melalui pemodelan *Data Flow Diagram* (DFD)

3. Kebutuhan Keluaran (*Output*)

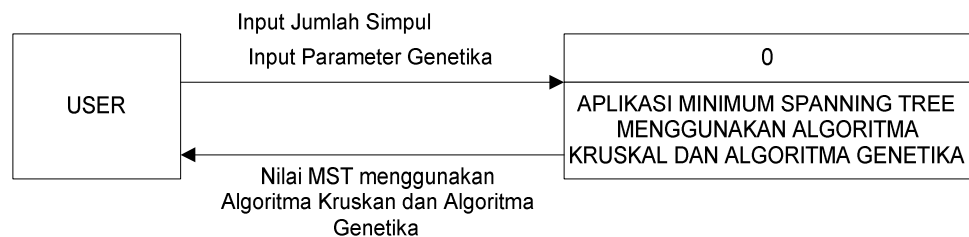
Keluaran dari sistem yang dibangun adalah nilai bobot minimum dari *Spanning Tree* dengan menggunakan Algoritma Kruskal dan Algoritma Genetika. Hasil Akhir juga berupa grafik *Minimum Spanning Tree* dengan menggunakan Algoritma Kruskal dan Algoritma Genetika.

4.2 Perancangan Sistem

4.2.1 Perancangan *Data Flow Diagram* (DFD)

4.2.1.1 *Context Diagram*

Sebelum merancang DFD sistem secara terinci, maka dibutuhkan suatu gambaran sistem yang dapat merepresentasikan sistem secara umum. Gambaran tersebut dapat dilihat pada *Context Diagram* atau Diagram Konteks berikut :

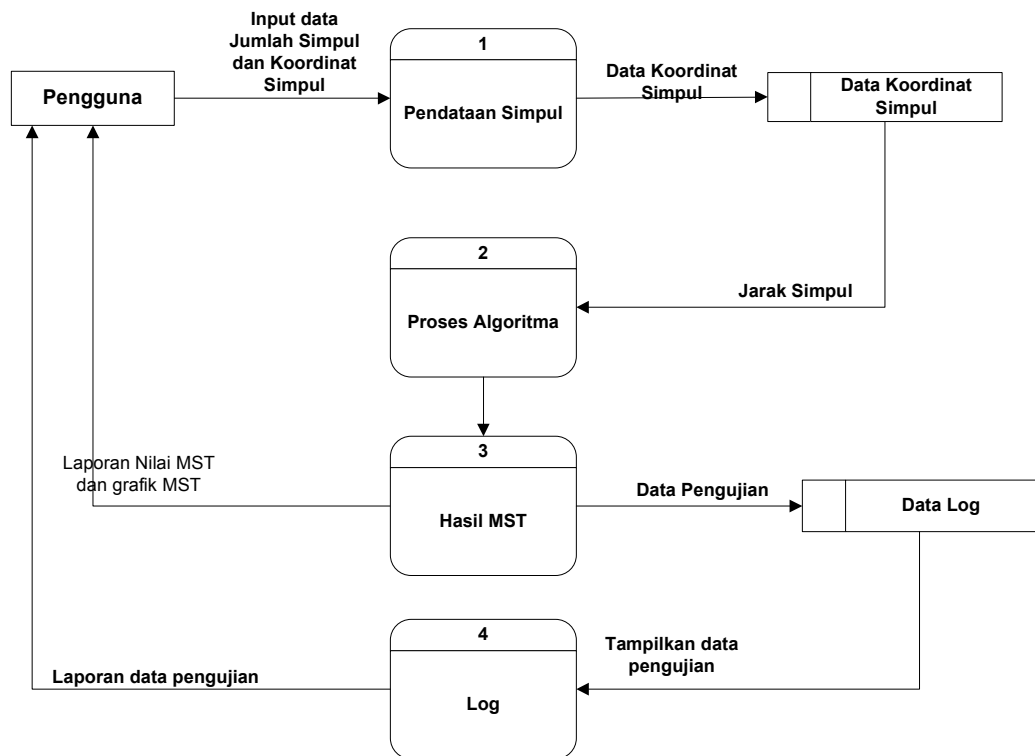


Gambar 4.1 Diagram Konteks Sistem

Gambar 4.1 Mengambarkan bahwa sistem terbentuk dari satu sistem besar yaitu Sistem Aplikasi *Minimum Spanning Tree* Menggunakan Algoritma Kruskal Dan Algoritma Genetika. Sistem dimulai dari *user* menginputkan Jumlah simpul dan Parameter Genetika kedalam sistem, yang kemudian sistem akan memberikan hasil *Minimum Spanning Tree* menggunakan Algoritma Kruskal dan Algoritma Genetika.

4.2.1.2 *DFD (Data Flow Diagram) Level- 0*

Diagram konteks kemudian dapat dikembangkan secara lebih terinci menjadi DFD Level-0. Proses-proses pada Level-0 merupakan dekomposisi dari proses utama Sistem Aplikasi *Minimum Spanning Tree* Menggunakan Algoritma Kruskal Dan Algoritma Genetika. DFD Level-0 dapat dilihat pada Gambar 4.2.

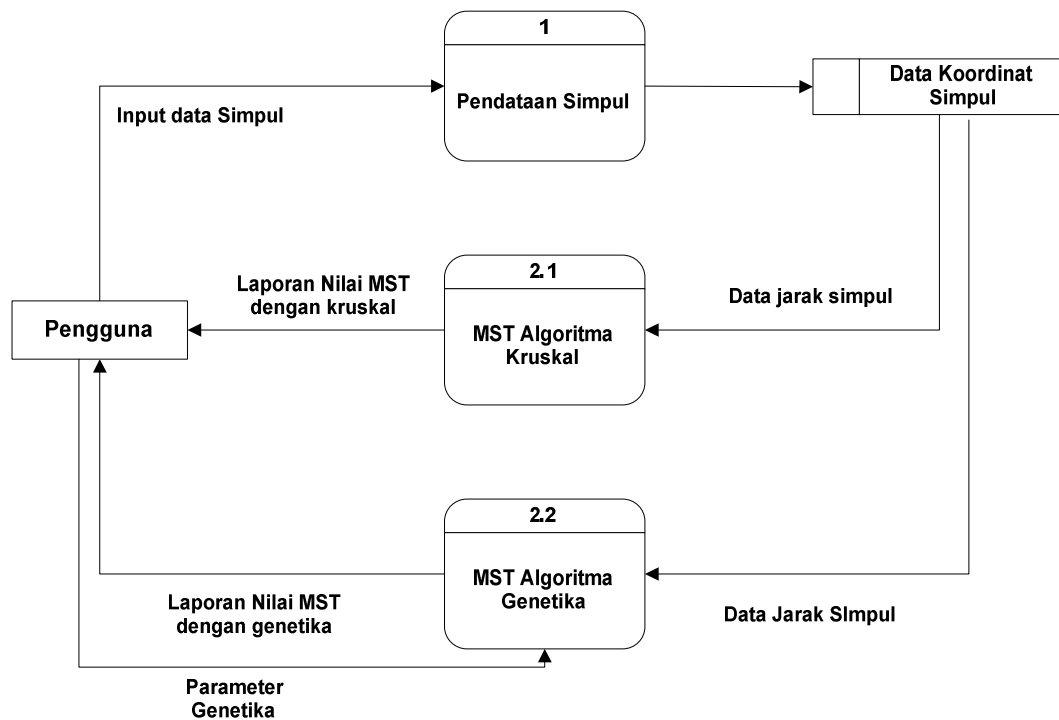


Gambar 4. 2 DFD Level-0

Gambar 4.2 menggambarkan 3 tahapan utama sistem. Tahapan pertama dimulai dengan pengguna menginputkan data jumlah simpul dan koordinat simpul yang kemudian akan disimpan di database koordinat simpul. Selanjutnya data simpul yang telah ada, data koordinat simpul didapatkan jarak antar simpul yang akan diproses pada tahapan proses algoritma. Tahapan proses algoritma memberikan hasil akhir berupa hasil bobot minimum MST dan dilaporkan kepada pengguna.

4.2.1.3 DFD (Data Flow Diagram) Level- 1

Data flow diagram level- 1 merupakan perincian dari DFD level- 0 proses 2 dan menjelaskan aliran dalam tahapan proses algoritma.

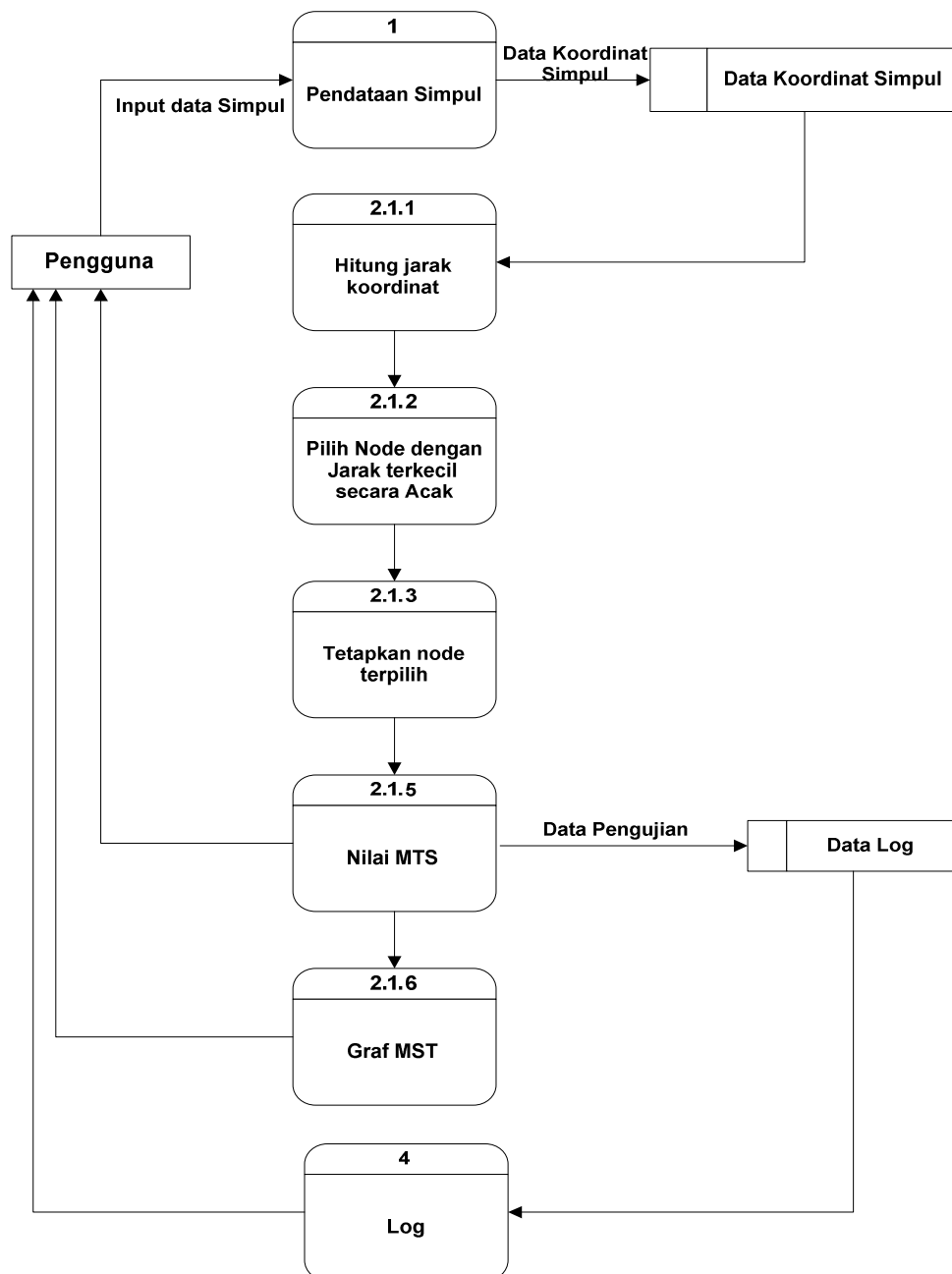


Gambar 4.3 DFD Level- 1

Dari Gambar 4.3, pada DFD level- 1 terdapat 2 tahapan proses yaitu *Minimum Spanning Tree menggunakan Algoritma Kruskal* dan *Minimum Spanning Tree menggunakan Algoritma Genetika*. Data dimulai setelah pengguna meng-*input*-kan koordinat simpul dan disimpan pada Data koordinat simpul. Data pada koordinat simpul kemudian dihitung nilai jarak antar simpul dan diproses menggunakan Algoritma Kruskal untuk menghasilkan nilai MST. Dan pada tahapan Genetika, Pengguna diharuskan kembali meng-*input*-kan nilai parameter-parameter algoritma Genetika dan juga data pada koordinat Simpul dihitung dan di proses ke data Edge untuk selanjutnya dapat digunakan untuk menghitung MST dengan Algoritma genetika. Hasil akhirnya kembali dilaporkan ke pengguna

4.2.1.4 DFD (Data Flow Diagram) Level- 2 Proses 1 (Minimum Spanning Tree dengan Algoritma Kruskal)

Data flow Diagram Level- 2 proses 1 merupakan perincian dari proses MST menggunakan algoritma kruskal



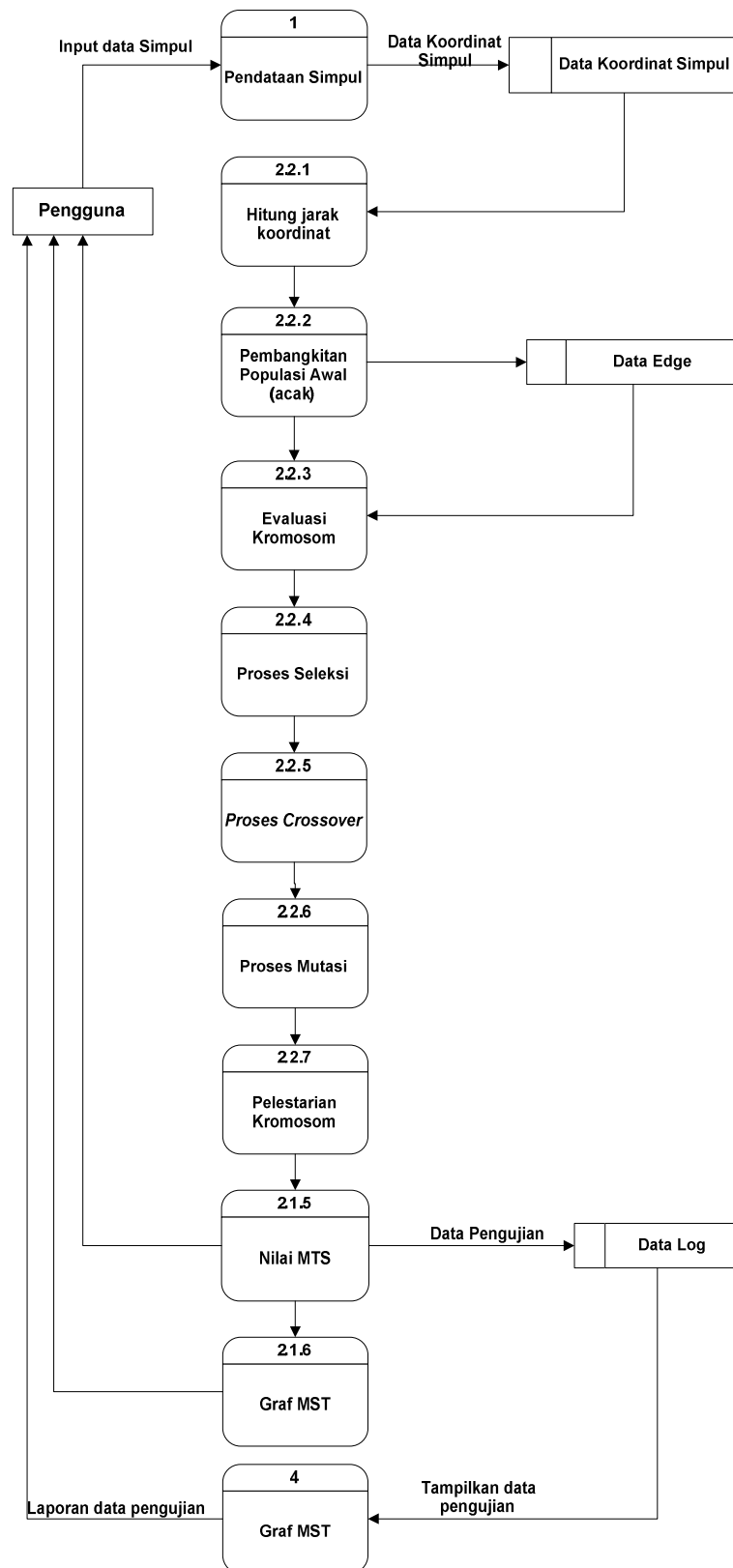
Gambar 4.4 DFD Level- 2 Proses 1, Penyelesaian MTS dengan Algoritma Kruskal

Pada Gambar 4.4. DFD Level-2 Proses 1, mrnggambarkan proses penyelesaian MTS dengan Algoritma Kruskal. Setelah data koordinat di-*input*-kan, selanjutnya dihitung jarak antar koordinat. Jarak yang paling kecil dari seluruh graf dipilih dan ditetapkan sebagai satu bagian tree. Ulangi langkah ini sampai seluruh graf terhubung. Dan untuk selanjutnya hitung bobot minimum dan gambar dari *tree*.

4.2.1.5 DFD (Data Flow Diagram) Level- 2 Proses 2 (Minimum Spanning Tree dengan Algoritma Genetika

Data flow Diagram Level- 2 proses 2 merupakan perincian dari proses MST menggunakan algoritma genetika. Tahapan pertama yang dilakukan setelah data koordinat di-*input*-kan adala menghitung jarak antar titik koordinat, kemudian membangkitkan populasi awal secara acak, mengevaluasi kromosom (menghitung bobot, nilai *fitness*, nilai probabilitas, dan nilai kumulatif), melakukan seleksi kromosom untuk dapat dijadikan kromosom induk pada *crossover*, melakukan persilangan/ rekombinasi, mutasi, dan melakukan pergantian kromosom yang dipilih sebagai bentuk pelestarian kromosom. Memilih hasil minimum dan menggambar *tree* yang terbentuk yang kemudian dilaporkan ke pengguna.

Data flow Diagram Level- 2 proses 2 dapat dilihat pada gambar 4.5 berikut.



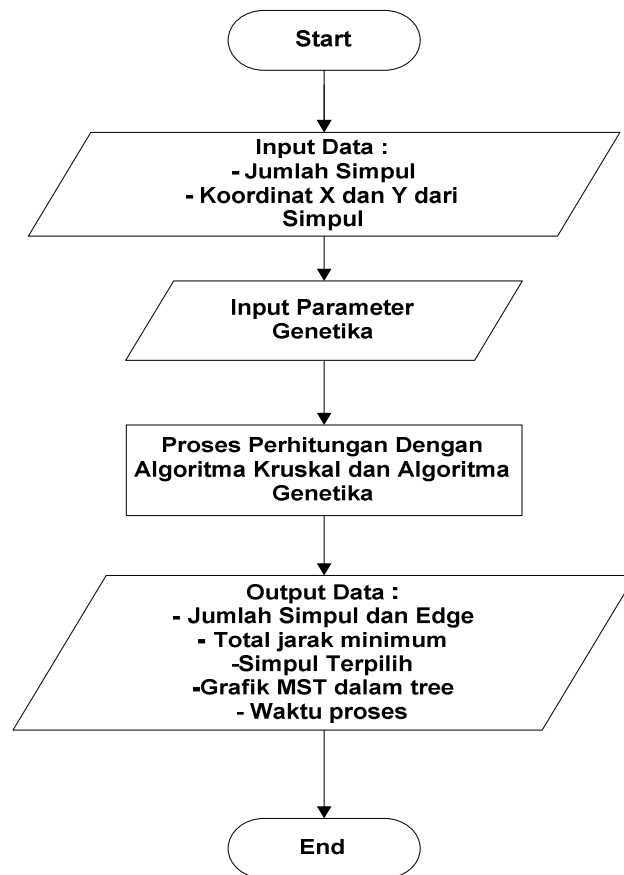
Gambar 4.5 DFD Level- 2 Proses 2, Minimum Spanning Tree dengan Algoritma Genetika

4.2.2 Perancangan *Flowchart* (Diagram Alir)

4.2.2.1 Perancangan *Flowchart* Sistem secara umum

Diagram alir sistem digunakan untuk menggambarkan keseluruhan langkah kerja sistem aplikasi yang akan dibuat. *Flowchart* sistem secara umum dalam penyelesaian *Minimum Spanning Tree* dapat dilihat pada Gambar 4.6.

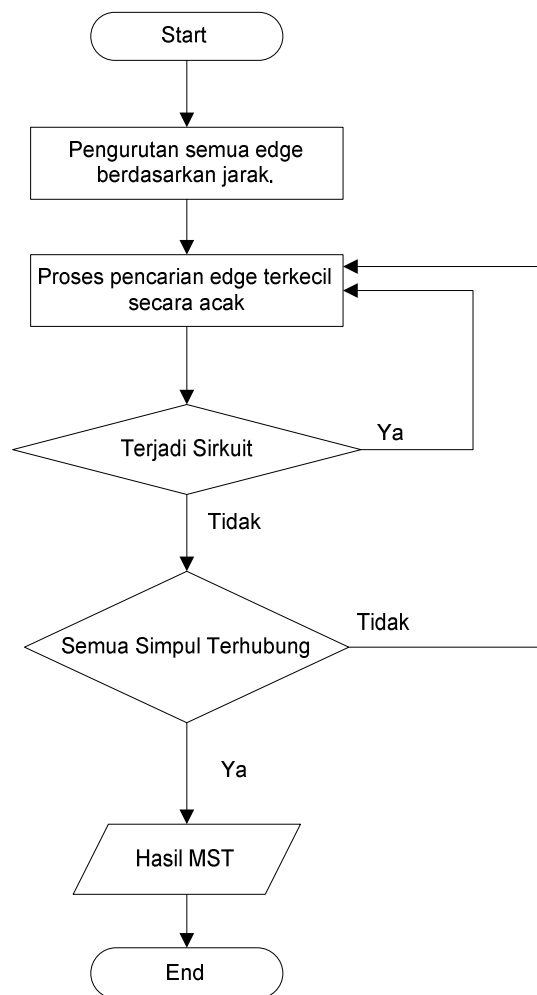
Sistem dimulai dengan menampilkan form antarmuka kepada pengguna. Form selanjutnya pengguna diharuskan mengisi data yang dibutuhkan, pengguna dapat menghapus data yang lama dan memulai dengan data baru, menambah data yang telah ada, atau dengan data lama tanpa penambahan atau pengurangan. Kemudian sistem akan memproses data tersebut dengan algoritma Kruskal, Sedangkan untuk algoritma genetika, pengguna diharuskan memasukkan parameter genetika. Setelah proses selesai, maka pengguna akan mendapatkan hasil nilai/ bobot minimum dari *Spanning tree*, Simpul dan edge terpilih, waktu proses dan gambaran *Minimum spanning tree* dalam bentuk graf.



Gambar 4. 6 Flowchart Sistem Aplikasi Secara Umum

4.2.2.2 Perancangan *Flowchart* Algoritma Kruskal

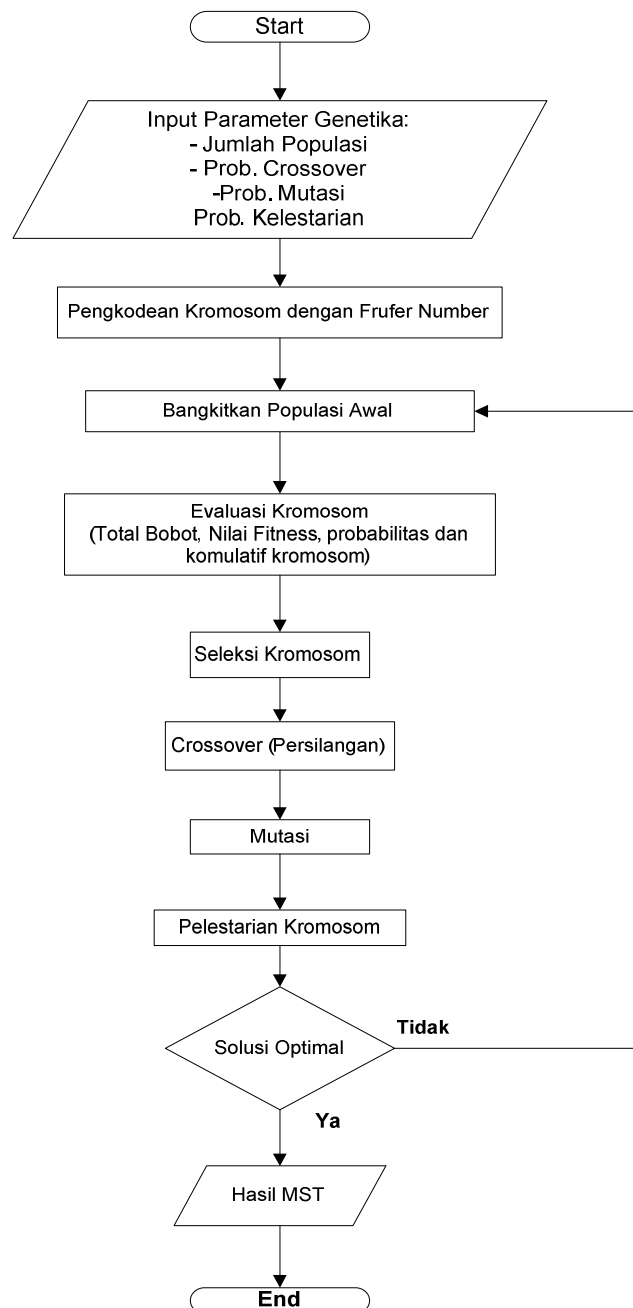
Flowchart Algoritma Kruskal yang akan diterapkan pada sistem dapat dilihat pada Gambar 4.7. *Flowchart* algoritma ini mengikuti langkah-langkah algoritma yang telah dijelaskan pada Sub bab 2.4



Gambar 4.7 Flowchart Algoritma Kruskal

4.2.2.3 Perancangan *Flowchart* Algoritma Genetika

Flowchart algoritma ini mengikuti langkah-langkah algoritma yang telah dijelaskan pada Sub bab 2.5. *Flowchart* Algoritma Genetika pada gambar 4.8 dibawah ini :



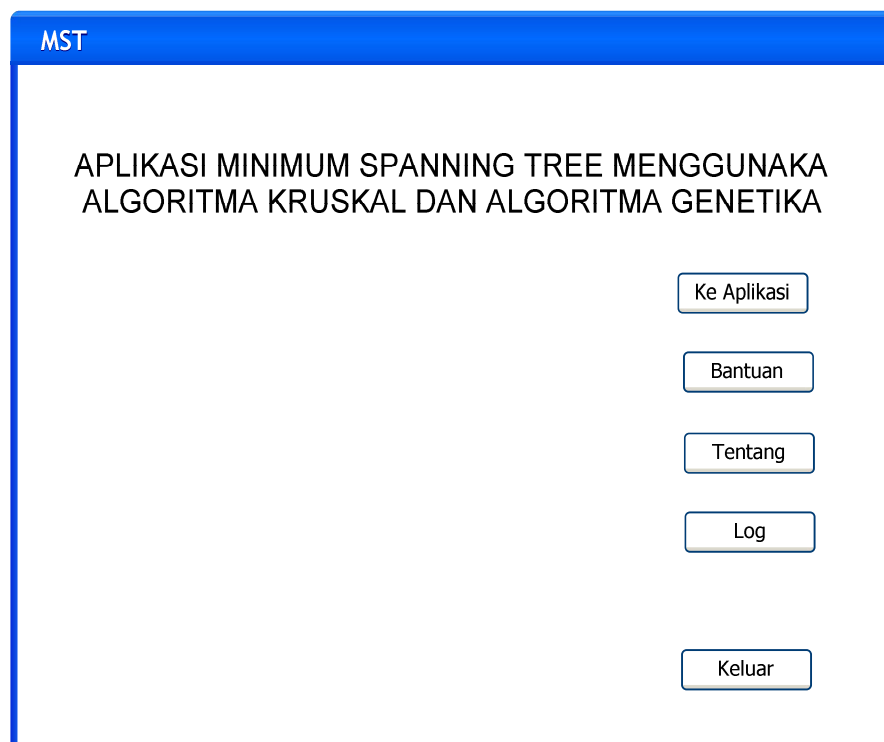
Gambar 4.8 Flowchart Algoritma genetika

4.2.3 Perancangan Antarmuka

Tujuan utama perancangan antarmuka (*interface*) adalah untuk mempermudah manusia (*user*) dalam mengoperasikan sistem sekaligus mendapatkan berbagai umpan balik yang diperlukan selama pengoperasiannya sehingga sistem dapat bekerja sesuai harapan.

4.2.3.1 Rancangan Antarmuka Judul

Bagian Antarmuka Judul menampilkan beberapa *Command button* yang berfungsi untuk menghubungkan ke halaman selanjutnya. Bagian ini dapat dilihat pada Gambar 4.9



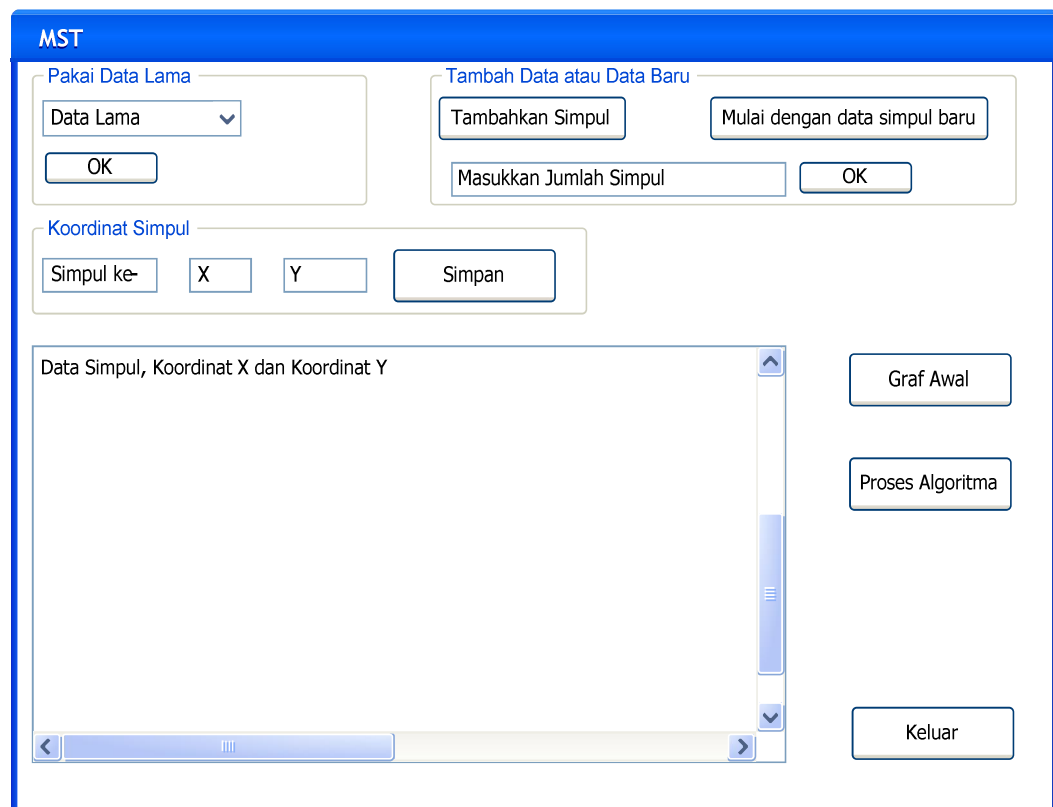
Gambar 4.9 Rancangan Antarmuka Judul

Bagian judul ini akan tampil di awal sistem dijalankan atau dapat disebut sebagai tampilan awal. Pengguna dapat masuk ke dalam menu utama dengan menekan tombol *Ke Aplikasi*, tombol *Bantuan* jika pengguna ingin mengetahui *manual* pemakaian aplikasi, tombol *tentang* berisikan informasi penulis, tombol

log berisi riwayat percobaan yang telah dilakukan dan tombol *keluar* jika pengguna akan menutup program.

4.2.3.2 Rancangan Antarmuka Input Koordinat

Bagian Antarmuka input koordinat, dimana pengguna *menginputkan* jumlah simpul dan koordinat simpul kedalam sistem, yang kemudian akan di proses dengan Algoritma Kruskal dan Algoritma Genetika. Antarmuka Input Koordinat pada Gambar 4.10



Gambar 4.10 Tampilan form input koordinat

Pada tampilan tersebut, terdapat 3 panel yaitu panel Pilihan Data, Jumlah Simpul dan Koordinat Simpul. Tabel Data Koordinat dan Tombol Graft awal, Proses Algoritma dan Keluar.

1. Panel Data Lama

Pada panel data lama, pengguna dapat memilih data koordinat simpul yang tersimpan di database. Data koordinat simpul ini adalah data yang telah pernah digunakan pada pengujian sebelumnya.

2. Panel Tambah Data atau Data Baru

Pada panel ini, pengguna dapat memilih menambahkan data koordinat simpul dari data yang ada sebelumnya dengan memilih tombol tambah data, atau menggunakan data yang benar-benar baru. Untuk menambah data ataupun menggunakan data baru, pengguna harus mengisi jumlah simpul terlebih dahulu, dan selanjutnya menekan tombol OK untuk program dapat mengeksekusi total simpul yang diinginkan

3. Panel Koordinat Simpul

Pada Panel ini, pengguna memasukkan data posisi Simpul, Koordinat X dan Koordinat Y dari simpul tersebut. Panel Koordinat Simpul ini dipengaruhi oleh Jumlah Simpul, sehingga *input* koordinat simpul belum bisa dilakukan jika pengguna belum memproses jumlah simpul, dan juga jumlah simpul yang dimasukkan hanya sebanyak yang diisikan pada panel Jumlah Simpul

4. Tabel Data Simpul

Merupakan daftar/ Tabel Simpul beserta koordinat X dan Koordinat Y dari simpul yang dimasukkan pada panel Koordinat Simpul Sebelumnya.

5. Tombol Graf Awal

Yaitu untuk pengguna melihat Graf awal dari Simpul-simpul yang dimasukkan sebelumnya.

6. Tombol Proses Algoritma

Ketika seluruh simpul dan koordinatnya dimasukkan, pengguna selanjutnya akan melakukan proses perhitungan MST dengan Algoritma Kruskal dan Genetika. Untuk itu pengguna meng-klik tombol Proses Algoritma untuk proses selanjutnya

7. Tombol keluar

Jika pengguna akan menutup program, dan pengguna kembali ke halaman judul

4.2.3.3 Rancangan Antarmuka Proses

The screenshot displays the 'MST' application window. At the top, there's a blue title bar with the text 'MST'. Below it, a section titled 'Input Parameter Genetika' contains four input fields: 'Jumlah Populasi', 'Prob. Crossover', 'Prob. Mutasi', and 'Prob. Kelestarian'. Below these fields are two buttons: 'Proses Kruskal' and 'Proses Genetika'. Under 'Proses Kruskal' is a text area labeled 'Hasil Algoritma Kruskal' with a vertical scrollbar. Similarly, under 'Proses Genetika' is a text area labeled 'Hasil Algoritma Genetika' with a vertical scrollbar. At the bottom of the window are three buttons: 'Keluar', 'Matrik Jarak', and 'Lihat Graf'.

Gambar 4.11 Rancangan Antarmuka Proses

Dari antarmuka *form* proses gambar 4.11, terdapat 1 panel input parameter Algoritma Genetika dan 2 papan informasi.

1. Panel Input Parameter Genetika

Pada panel ini, pengguna diminta memasukkan parameter-parameter algoritma genetika yaitu jumlah populasi, peluang *crossover*, peluang mutasi dan kelestarian kromosom. *Input* Jumlah populasi bernilai bilangan bulat positif, sedangkan peluang *crossover*, peluang mutasi dan kelestarian kromosom yaitu 0 sampai 1

2. Proses Kruskal

Tombol Proses Kruskal adalah untuk melakukan perhitungan dengan menggunakan Algoritma Kruskal

3. Proses Algoritma Genetika

Tombol Proses Genetika adalah untuk melakukan perhitungan dengan menggunakan Algoritma Genetika

4. Informasi

Berupa hasil akhir MST dengan menggunakan Algoritma Kruskal dan Algoritma Genetika.

5. Lihat Graf

Jika Pengguna ingin melihat hasil MST dalam bentuk graf dan grafik tree nya

6. Matriks Jarak

Yaitu berupa tampilan matrik jarak antar simpul dari graf awal

7. Keluar

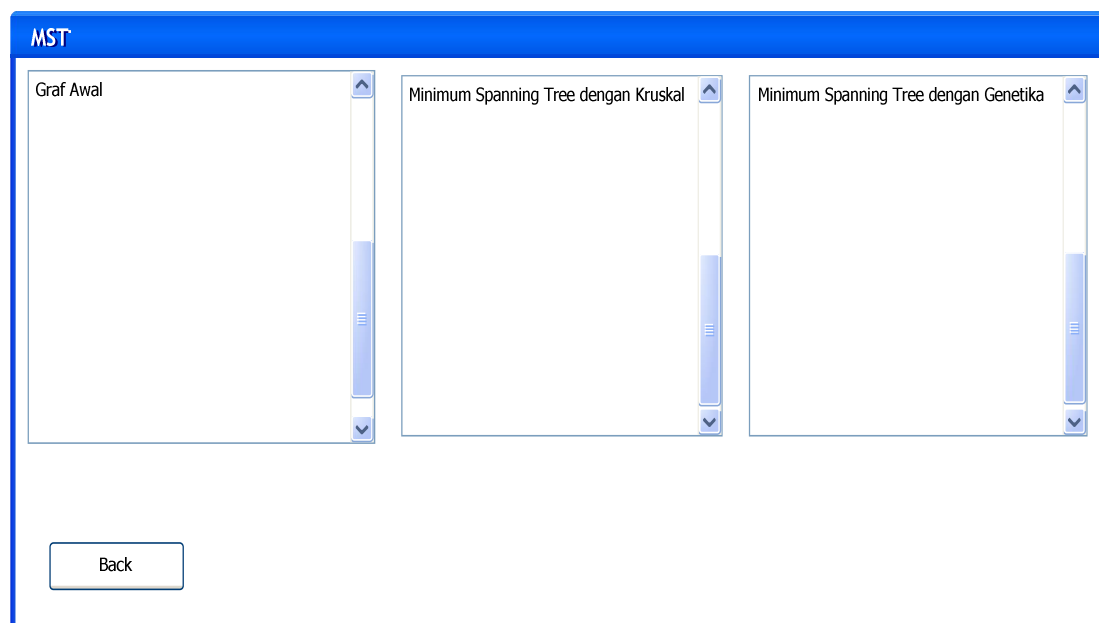
Jika pengguna akan menutup program, dan pengguna kembali ke halaman judul

4.2.3.4 Rancangan Antarmuka Tampilan Graf

Pada Tampilan graf, yaitu jika pengguna ingin melihat hasil MST dari data yang telah di inputkan dalam bentuk grafik tree nya. Pada halaman ini akan menampilkan 3 grafik, yaitu

1. Gambar Graf awal
2. Gambar *Minimum Spanning Tree* dengan Algoritma Kruskal
3. Gambar *Minimum Spanning Tree* dengan Algoritma Genetika

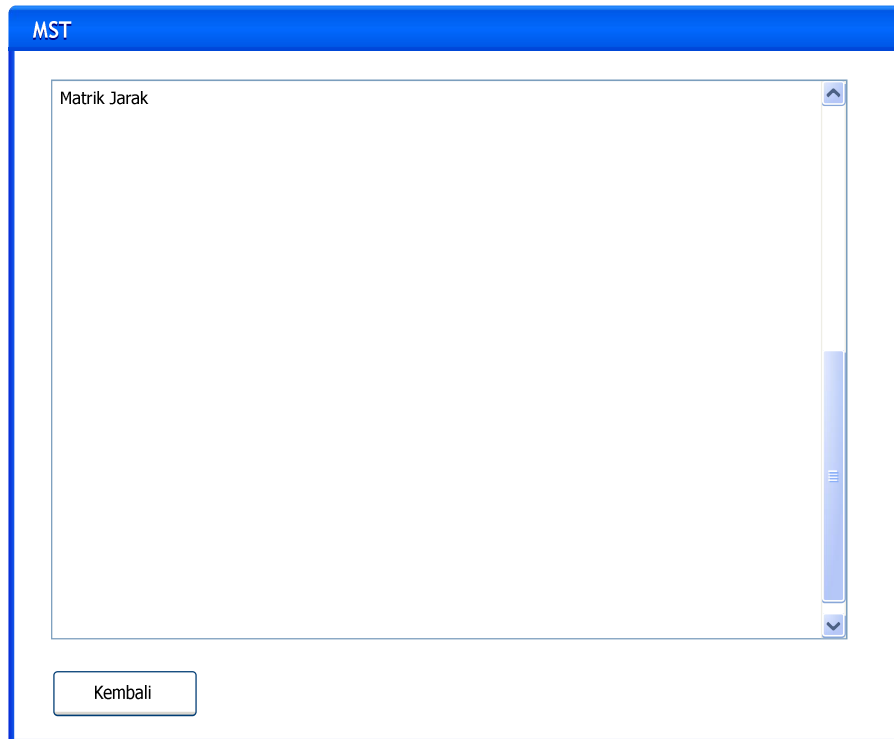
Rancangan antarmuka tampilan graf dapat dilihat pada gambar 4.12 :



Gambar 4.12 Antarmuka Tampilan Graf

4.2.3.5 Rancangan Antarmuka Tampilan Matriks Jarak

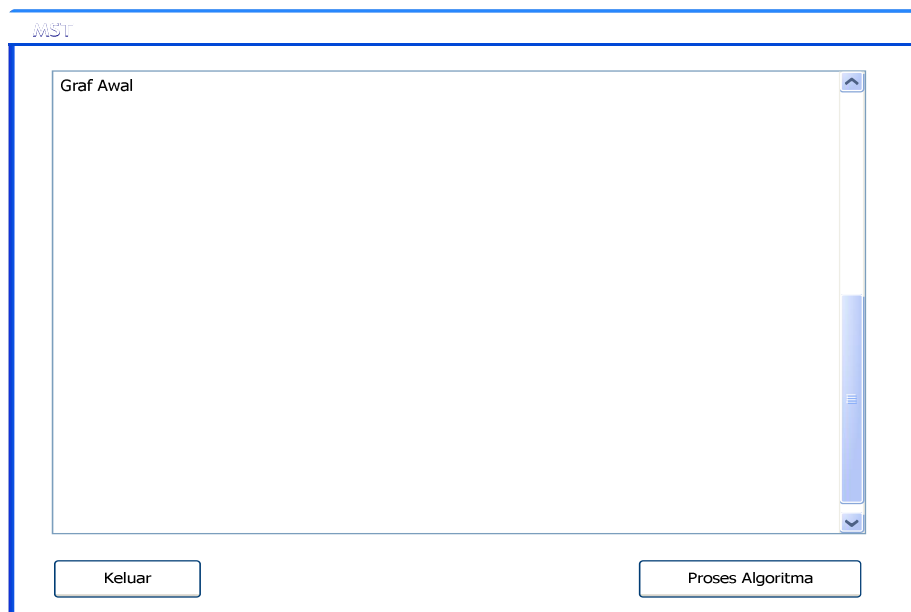
Tampilan matriks jarak yaitu menampilkan data jarak awal antar titik simpul dari graf awal. Rancangan antarmuka halaman Matrik Jarak Gambar 4.13



Gambar 4. 13 Antarmuka tampilan halaman Matrik Jarak

4.2.3.6 Rancangan Antarmuka Tampilan Graf Awal

Tampilan graf awal yaitu menampilkan grafik awal dari koordinat yang di *input*-kan. . Rancangan antarmuka halaman Graf Awal Gambar 4.14.

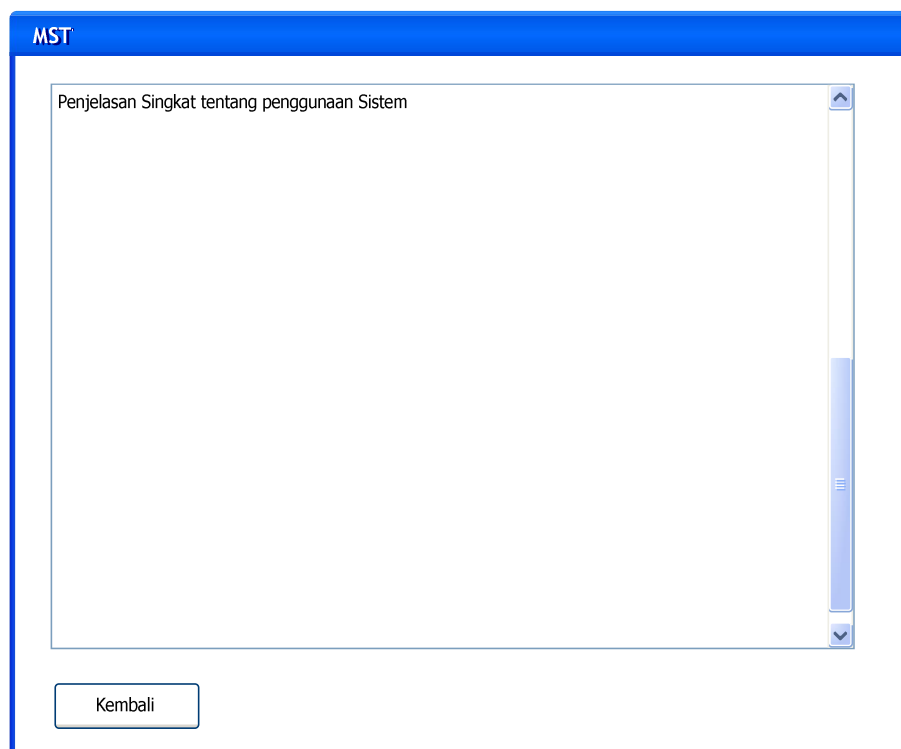


Gambar 4.14 Antarmuka Tampilan Graf Awal

Pada halaman ini, hanya menampilkan gambar dari graf awal lengkap yang merupakan bentuk hasil dari masukan data Simpul, Koordinat X dan Koordinat Y yang di *input*-kan sebelumnya pada halaman Koordinat.

4.2.4.4 Rancangan Antarmuka Bantuan

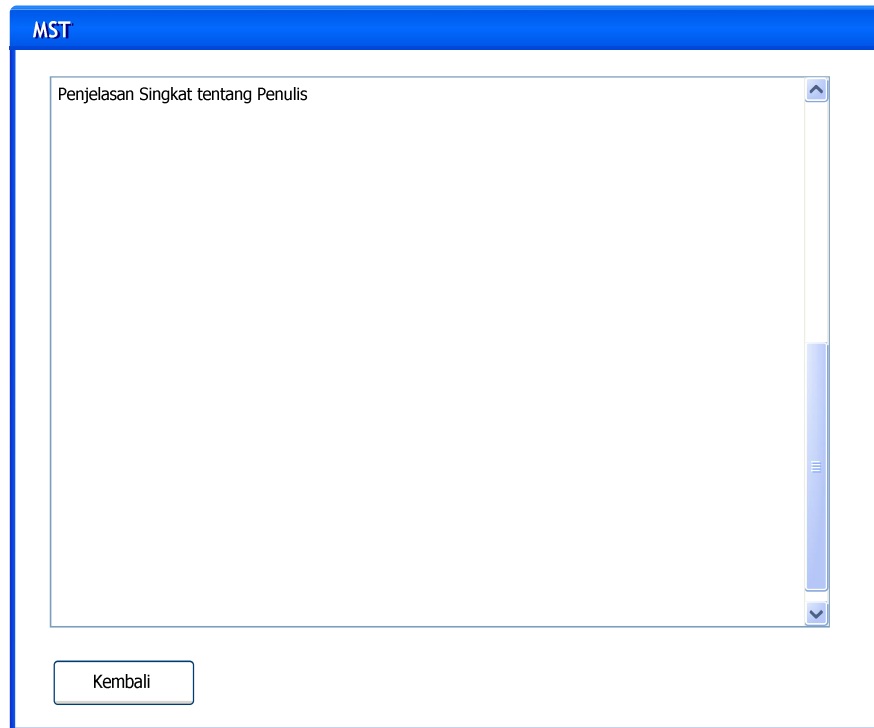
Yaitu berisi penjelasan penggunaan sistem secara manual. Rancangan antarmuka halaman bantuan Gambar 4.15.



Gambar 4.15 Antarmuka Bantuan

4.2.3.7 Rancangan Antarmuka Tentang

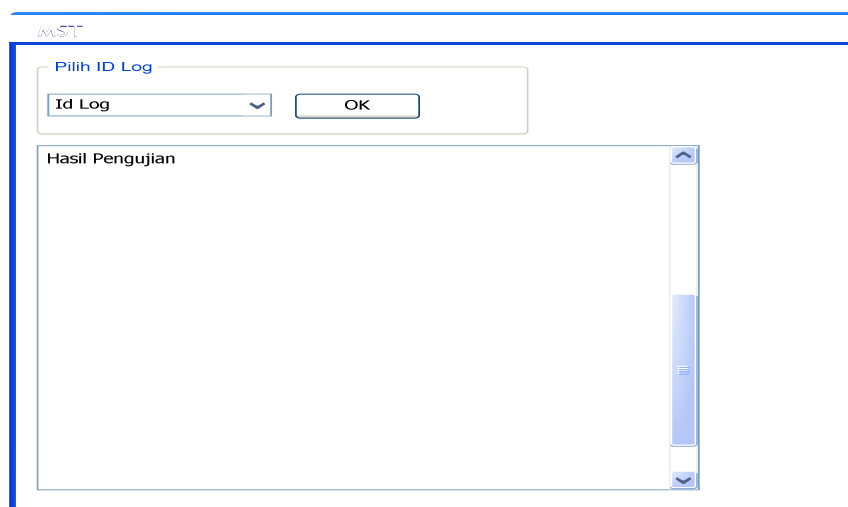
Berisi penjelasan tentang penulis. Rancangan antarmuka halaman Bantuan Gambar 4.16



Gambar 4.16 Antarmuka Tentang

4.2.3.8 Rancangan Antarmuka Log

Pada Halaman ini, pengguna harus memilih pengujian yang akan dilihat hasilnya. Daftar pengujian ini dapat dilihat pada panel pilih ID log, dan tekan tombol OK, maka akan ditampilkan hasil pengujian dari data uji yang dipilih. Antar muka halaman Log bisa dilihat pada Gambar 4.17



Gambar 4.17 Antarmuka Log

4.2.4 Perancangan Basis Data

4.2.4.1 Perancangan Tabel

Basis Data terdiri dari tabel-tabel yang digunakan untuk menyimpan data yang diperlukan dalam sistem. Tabel-tabel yang digunakan dalam sistem adalah sebagai berikut :

1. Tabel Koordinat Percobaan

Tabel ini menyimpan data koordinat dari percobaan yang dilakukan oleh pengguna. Tiap percobaan yang, data koordinat yang digunakan akan disimpan dalam tabel Detail Koordinat Percobaan

Tabel 4.1 Rancangan Tabel Detail Koordinat Percobaan

Kolom	Tipe
Id_Percobaan	Varchar
Simpul	Int
X	Double
Y	Double

2. Tabel Edge

Tabel ini menyimpan *record* sejumlah edge terpilih yang kemudian akan digunakan untuk perhitungan *Minimum Spanning Tree*.

Tabel 4.2 Rancangan Tabel Edge

Kolom	Tipe
ID	Int
Edge Kiri	Double
Edge Kanan	Double
Bobot	Double

3. Tabel Log

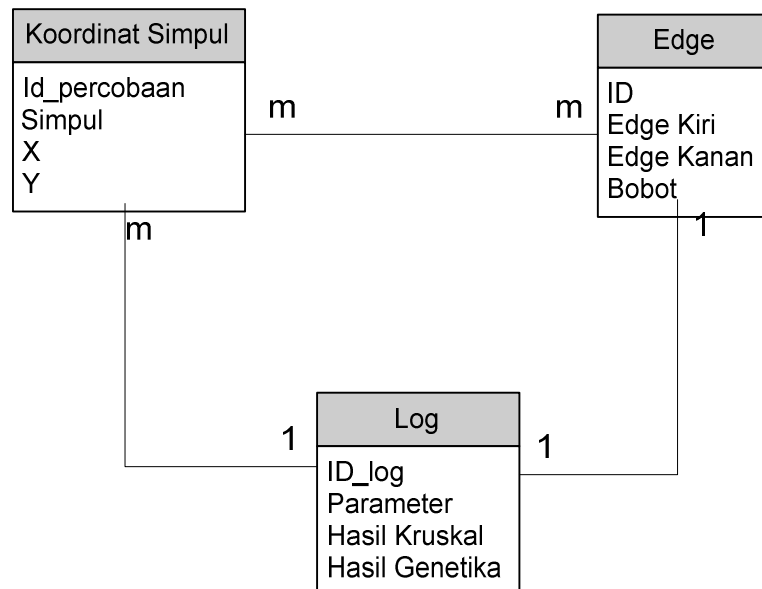
Tabel ini menyimpan *record* percobaan yang dilakukan

Tabel 4.3 Rancangan Tabel Log

Kolom	Tipe
ID_log	Varchar
Parameter	Double
Hasil Kruskal	Double
Hasil Genetika	Double

4.2.4.2 Perancangan *Entity Relation Diagram* (ERD)

Pada penelitian ini, database yang dibangun saling berhubungan dan berketerkaitan satu sama lain. Hubungan antar tabel dalam aplikasi ini dapat dilihat pada gambar *Entity Relation Diagram* (ERD) berikut ini :



BAB V

HASIL DAN PEMBAHASAN

5.1 Implementasi Sistem

Implementasi merupakan tahap dimana sistem siap dioperasikan pada tahapan yang sebenarnya, sehingga akan diketahui apakah sistem yang telah dibuat benar-benar sesuai dengan yang direncanakan. Tahap implementasi merupakan lanjutan dari tahap pembahasan dan perancangan.

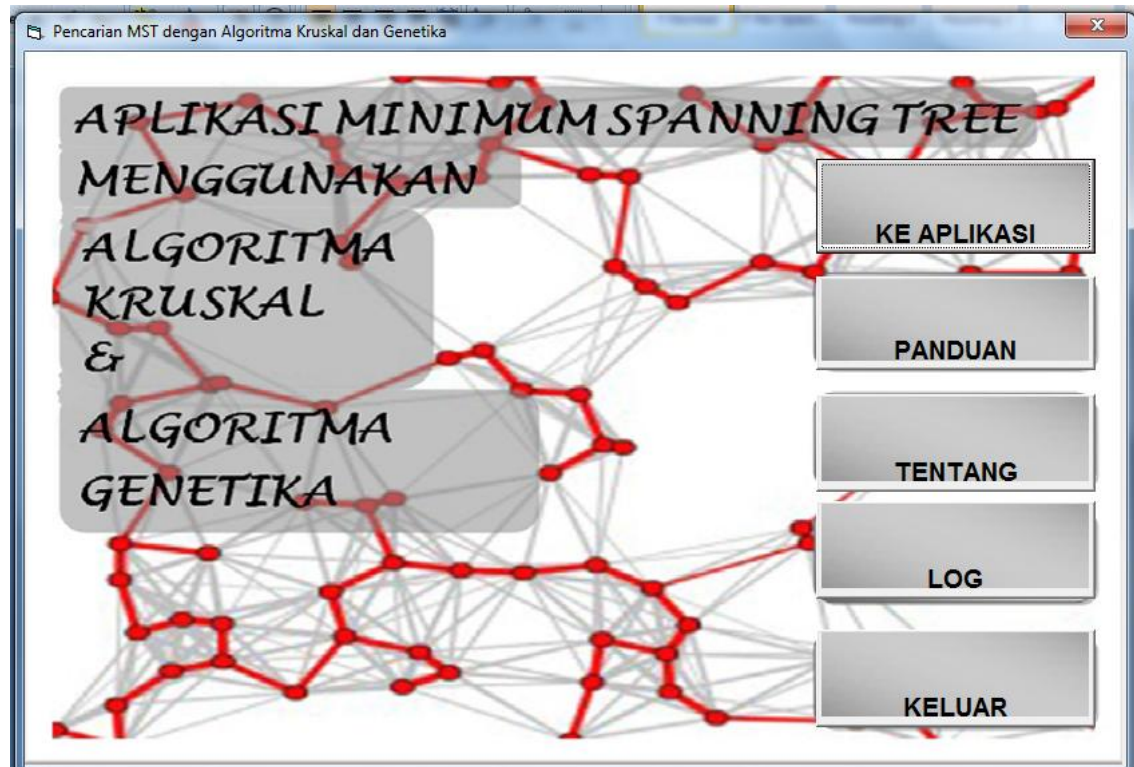
Pada tahapan implementasi sistem Aplikasi Minimum Spanning Tree menggunakan Algoritma Kruskal dan Algoritma Genetika, adalah menerapkan rancangan dan desain antarmuka yang telah di rancang dan dijelaskan pada subbab 4.2.3. Pada tahap ini, sistem diimplementasikan dengan bahasa pemrograman Visual Basic 6.0 dan database MySQL dengan koneksi Mysql-*connector-odbc5.1.13-win32*. Implementasi Sistem terbagi atas dua yaitu Implementasi Antarmuka dan Implementasi prosedur. Berikut adalah hasil implementasi dari sistem.

5.1.1 Implementasi Antar Muka

Berikut adalah hasil implementasi antar muka dari sistem yang dibangun :

A. Tampilan Halaman Utama

Antarmuka awal dari aplikasi *Minimum Spanning Tree* Menggunakan Algoritma Kruskal dan Algoritma genetika merupakan halaman antarmuka yang pertama kali muncul ketika aplikasi dibuka, dapat dilihat pada Gambar 5.1.



Gambar 5. 1 Hasil Implementasi halaman Utama

B. Tampilan Form Koordinat

Tampilan *Form* Koordinat akan muncul jika pilihan Ke Aplikasi pada halaman antarmuka utama diklik. Pada halaman koordinat, pengguna akan diminta di-*input*-kan data koordinat simpul yang akan di hitung. *Input* koordinat bisa dengan menggunakan data yang telah di uji sebelumnya, atau menggunakan data baru untuk selanjutnya dapat diproses.

Implementasi dari rancangan antarmuka input koordinat dapat dilihat pada gambar 5.2

Menentukan Koordinat Simpul

Pakai data lama: P00528

OK

Tambah Data atau Data Baru

Tambahkan Simpul Mulai dengan Data Simpul Baru

Masukkan Jumlah Simpul : OK

Input Koordinat Simpul

Simpul: Titik X Titik Y Simpan

Simpul	X	Y
1	5	5
2	5	10
3	10	20
4	15	25
5	20	10
6	30	25
7	20	15
8	5	30
9	10	30
10	30	30

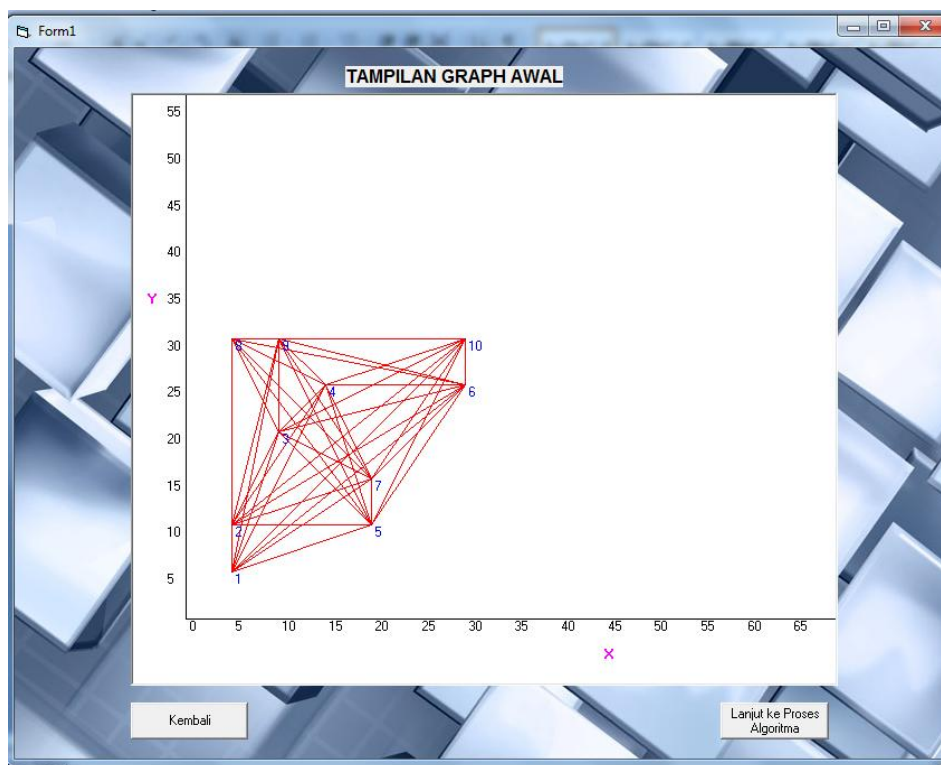
Tampilkan Graf

Proses Algoritma

Batal

Gambar 5.2 Hasil Implementasi Halaman Input data Koordinat

C. Tampilan Graf Awal



Gambar 5.3 Hasil Implementasi Tampilan Awal Graf

Gambar 5.3 merupakan implementasi dari rancangan antarmuka input koordinat. Tampilan *Form* Graf Awal akan muncul jika pilihan Tampilkan Graf pada halaman antarmuka input koordinat diklik. Pada halaman ini akan ditampilkan graf awal dalam koordinat kartesius dari data koordinat simpul yang telah di-*input*-kan sebelumnya. Graf yang ditampilkan adalah graf lengkap, berbobot, tidak berarah dan semua titik terhubung.

D. Tampilan Form Proses

Masukkan Parameter untuk Algoritma Genetika:

Populasi:	Prob. Crossover:	Prob. Mutasi:	Prob. Kelestarian:
50	0.2	0.2	1

Proses Kruskal

Hasil Kruskal:

Jumlah Edge Keseluruhan sebelumnya = 45
 Total Bobot Keseluruhan sebelumnya = 794.58
 Jumlah Edge MST = 9
 Total Bobot MST = 70.64
 Edge yang terpilih dari graf adalah :

1 - 2	= 5
2 - 3	= 11.18
3 - 4	= 7.07
3 - 7	= 11.18
4 - 9	= 7.07
5 - 7	= 5
6 - 7	= 14.14
6 - 10	= 5
8 - 9	= 5

Waktu Proses Algoritma Kruskal: 0.025524 detik

Proses Genetika

Hasil Genetika:

Jumlah Edge Keseluruhan sebelumnya = 45
 Total Bobot Keseluruhan sebelumnya = 794.58
 Jumlah Edge MST = 9
 Total Bobot MST = 94.3
 Edge yang terpilih dari graf adalah :

1 - 5	= 15.81
2 - 7	= 15.81
3 - 7	= 11.18
4 - 7	= 11.18
4 - 8	= 11.18
5 - 7	= 5
6 - 7	= 14.14
6 - 10	= 5
8 - 9	= 5

Waktu Proses Algoritma Genetika: 8.693116 detik

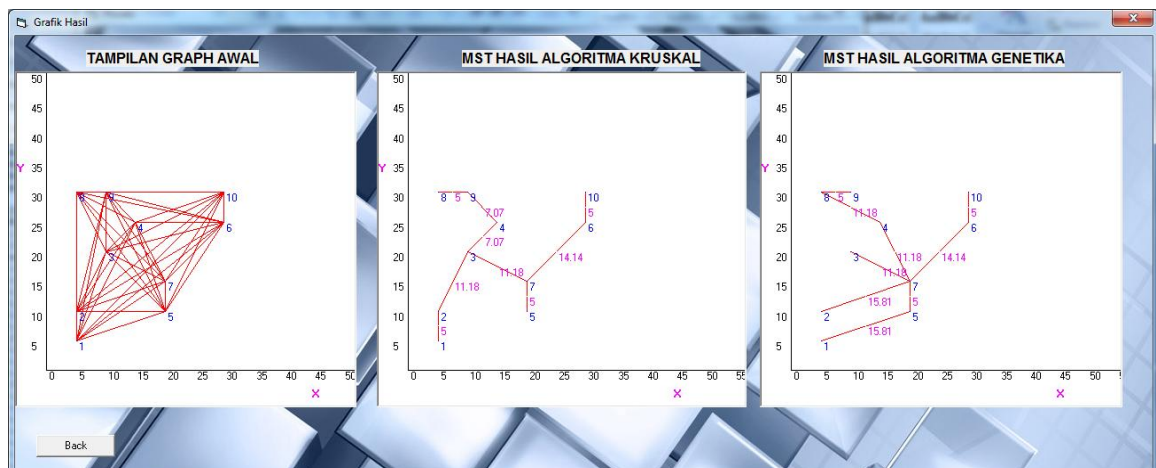
Kembali Lihat Data Jarak Grafik Hasil

Gambar 5.4 Hasil Implementasi Tampilan Proses Algoritma

Gambar 5.4 merupakan implementasi dari rancangan antarmuka proses. Tampilan *Form* proses akan muncul jika pilihan Proses Algoritma pada halaman antarmuka input koordinat diklik. Pada halaman ini akan ditampilkan hasil dari perhitungan MST dengan menggunakan Algoritma Kruskal dan Algoritma

Genetika. Namun sebelum itu pengguna harus meng-*input*-kan parameter-parameter Algoritma Genetika yang dibutuhkan dan kemudian klik tombol Proses Kruskal untuk memulai penghitungan menggunakan Algoritma Kruskal, dan klik tombol Proses Genetika untuk memulai penghitungan menggunakan Algoritma genetika. Kolom Hasil Kruskal dan Hasil Genetika akan menampilkan hasil MST dari data simpul yang telah di-*input*-kan pada halaman input koordinat sebelumnya.

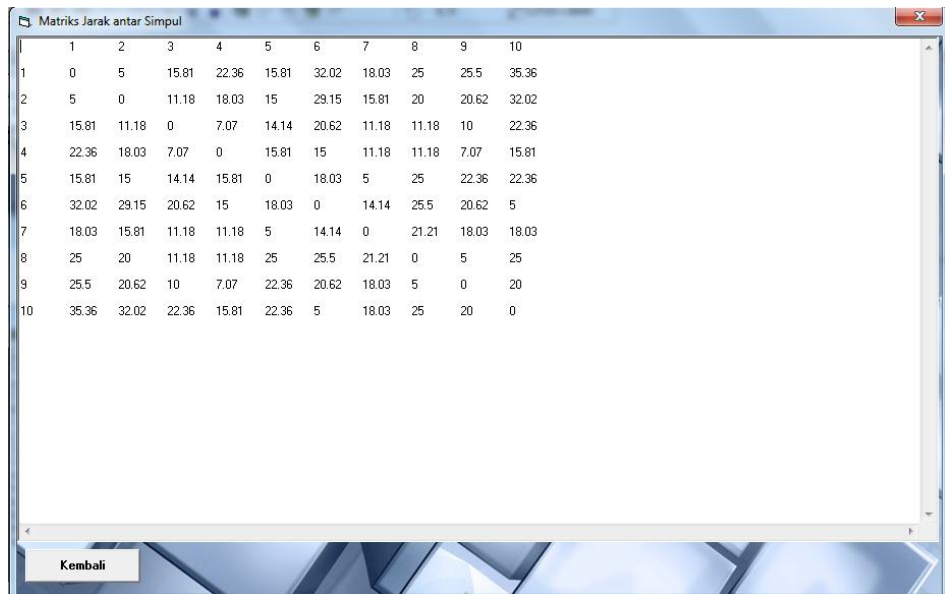
E. Tampilan From Lihat Grafik Hasil



Gambar 5.5 Hasil Implementasi Tampilan Grafik MST

Gambar 5.5 merupakan implementasi dari rancangan antarmuka lihat Grafik Hasil. Tampilan *Form* Grafik akan muncul jika pilihan Grafik Hasil pada halaman proses diklik. Pada halaman ini akan ditampilkan hasil dari perhitungan MST dengan menggunakan Algoritma Kruskal dan Algoritma Genetika dalam bentuk grafik. Grafik 1 menampilkan Graf awal, Grafik 2 merupakan edge terpilih beserta bobot nya untuk MST dari Algoritma Kruskal dan grafik 3 adalah merupakan edge terpilih beserta bobot nya untuk MST dari Algoritma Genetika.

F. Tampilan Halaman Matriks Jarak



	1	2	3	4	5	6	7	8	9	10
1	0	5	15.81	22.36	15.81	32.02	18.03	25	25.5	35.36
2	5	0	11.18	18.03	15	29.15	15.81	20	20.62	32.02
3	15.81	11.18	0	7.07	14.14	20.62	11.18	11.18	10	22.36
4	22.36	18.03	7.07	0	15.81	15	11.18	11.18	7.07	15.81
5	15.81	15	14.14	15.81	0	18.03	5	25	22.36	22.36
6	32.02	29.15	20.62	15	18.03	0	14.14	25.5	20.62	5
7	18.03	15.81	11.18	11.18	5	14.14	0	21.21	18.03	18.03
8	25	20	11.18	11.18	25	25.5	21.21	0	5	25
9	25.5	20.62	10	7.07	22.36	20.62	18.03	5	0	20
10	35.36	32.02	22.36	15.81	22.36	5	18.03	25	20	0

Gambar 5.6 Hasil Implementasi Halaman Matrik Jarak

Gambar 5.6, halaman Matrik jarak akan muncul jika pilihan matrik jarak pada halaman proses diklik. Pada halaman matrik jarak ini memperlihatkan jarak antar simpul dari graf awal.

G. Tampilan Menu Panduan

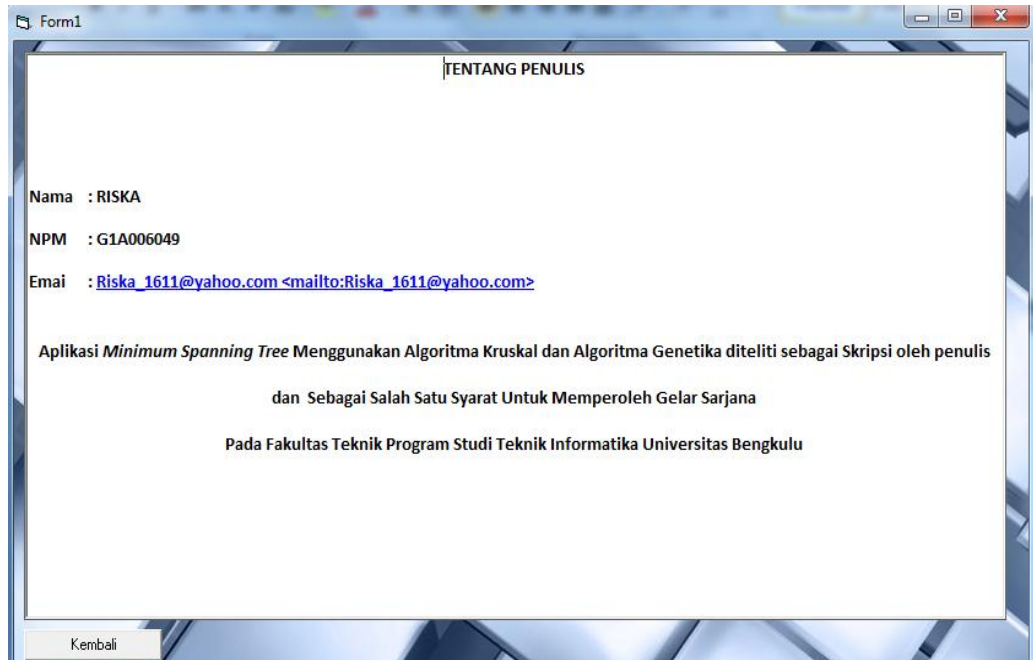
Tampilan *Form* Panduan akan muncul jika pilihan Panduan pada halaman antarmuka utama diklik. Pada halaman ini pengguna bisa membaca penggunaan sistem secara manual. Implementasi dari rancangan antarmuka panduan dapat dilihat pada gambar 5.7



Gambar 5.7 Hasil Implementasi Menu Panduan

H. Tampilan Menu Tentang

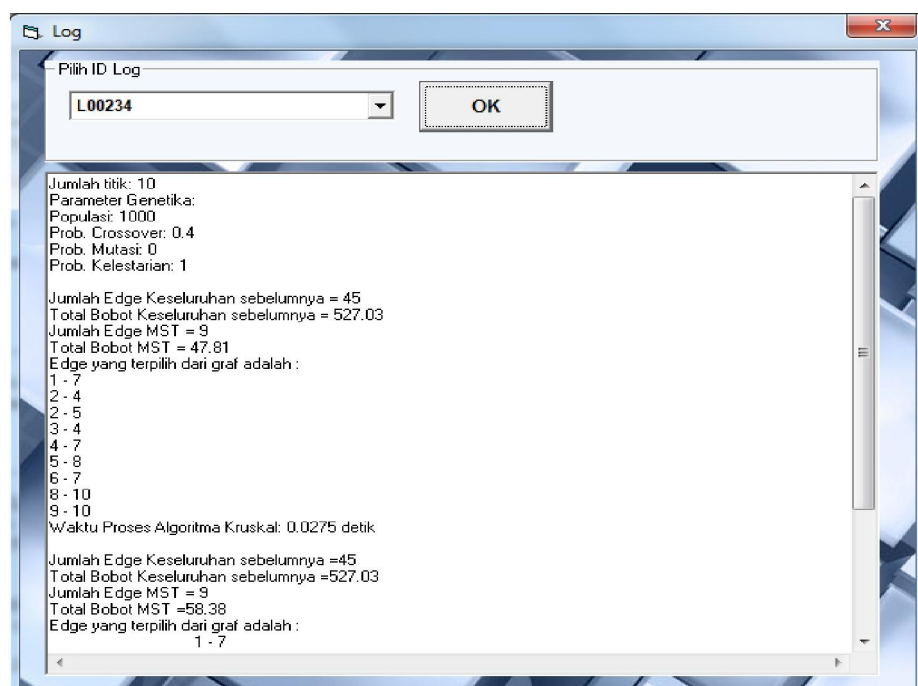
Tampilan *Form* Tentang akan muncul jika pilihan Tentang pada halaman antarmuka utama diklik. Pada halaman ini berisi data penulis. Implementasi dari rancangan antarmuka tentang dapat dilihat pada gambar 5.8



Gambar 5.8 Hasil Implementasi Menu Tentang

I. Tampilan Menu Log

Tampilan *Form* Log akan muncul jika pilihan Log pada halaman antarmuka utama diklik. Pada halaman ini berisikan hasil dari pengujian-pengujian yang telah dilakukan sebelumnya. Implementasi dari rancangan antarmuka log dapat dilihat pada gambar 5.9



Gambar 5.9 Hasil Implementasi Menu Log

5.1.2 Implementasi Prosedur

Implementasi prosedur merupakan implementasi prosedur dan perhitungan dari algoritma kruskal dan prosedur algoritma genetika yang menjadi inti dari perangkat aplikasi ini ke dalam bahasa pemrograman. Penerapan algoritma dalam program aplikasi ini diharapkan akan memberikan hasil *Minimum* dari *Spanning Tree* yang terbaik dan menghasilkan *Spanning Tree* dengan nilai paling minimum. Adapun hasil implementasi prosedur dapat dilihat pada Lampiran A laporan ini.

5.2 Pengujian Sistem

Proses pengujian sistem dilakukan secara keseluruhan terhadap sistem yang telah dibangun. Hasil dari pengujian ini, diharapkan dapat diketahui kekurangan-kekurangan dari sistem untuk kemudian dapat diperbaiki dan kesalahan dari sistem dapat diminimalisasi. Pengujian sistem ini dilakukan dengan dua cara yaitu pengujian *blackbox* menyinggung ujicoba yang dilakukan pada interface software dan pengujian metode *Sample Training* dengan pengecekan perubahan data dengan metode rancangan percobaan.

5.2.1 Pengujian Blackbox

Pengujian sistem dilakukan secara kompleks dan diharapkan dapat diketahui kekurangan-kekurangan dari sistem untuk kemudian dapat diperbaiki. Dengan demikian, kesalahan dari sistem dapat diminimalisasi atau bahkan dihilangkan. Pengujian sistem ini dapat dilakukan dengan mengisi data jumlah simpul, koordinat simpul dan parameter-parameter Algoritma Genetika.

Pengujian yang dilakukan adalah pengujian tidak normal, yaitu pengujian yang dilakukan dengan memberikan masukan dengan spesifikasi yang tidak diijinkan sehingga sistem akan memberikan reaksi lain. Reaksi sistem berupa berupa peringatan (*alert*) atau penanganan kesalahan (*error handling*).

- A. Peringatan jika jumlah simpul yang di isikan kecil dari 3.

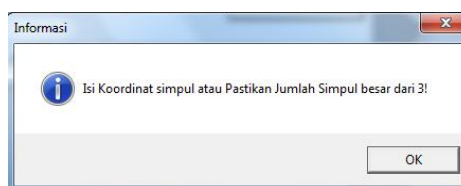
Bentuk peringatan dapat dilihat pada gambar 5.10 :



Gambar 5.10 Contoh Peringatan Jika data simpul yang diisikan kecil dari 3

- B. Peringatan jika pengguna tidak memasukkan data jumlah simpul terlebih dahulu namun akan menginputkan data koordinat.

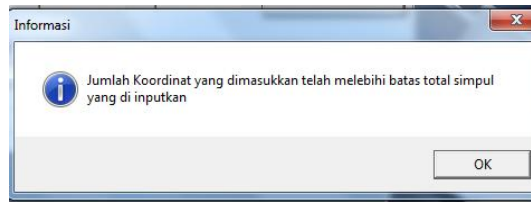
Bentuk peringatan dapat dilihat pada gambar 5.11 :



Gambar 5. 11 Contoh Peringatan Jika data simpul belum dimasukkan

- C. Peringatan jika data koordinat yang diinputkan lebih besar dari jumlah simpul yang telah ditentukan sebelumnya

Bentuk peringatan dapat dilihat pada gambar 5.12 :



Gambar 5.12 Contoh Peringatan Jika jumlah data simpul yang akan dimasukkan telah melebihi dari yang ditentukan sebelumnya

5.2.2 Pengujian *Sampel Testing*

Pada pengujian ini, dilakukan dengan metode *sample testing* dimana tiap percobaan memiliki perbedaan, yaitu dari jumlah simpul yang digunakan dan parameter-parameter genetika yang di inputkan. Pengujian dilakukan dengan menggunakan data yang di-*input*-kan secara acak oleh pengguna, data yang di-*input*-kan yaitu berupa titik koordinat X dan koordinat Y dari simpul, juga parameter-parameter Algoritma genetika. Parameter genetika yang digunakan yaitu jumlah populasi, peluang *crossover*, peluang Mutasi serta peluang kelestarian. Pada jumlah populasi genetika, jumlah maksimal populasi yang di inputkan adalah n^{n-2} dimana n adalah jumlah simpul. Sebagai contoh jika data simpul = 3, maka jumlah maksimal populasi yang dibangkitkan adalah sebanyak $3^{3-2} = 3^1 = 3$. Jika data populasi yang dibangkitkan lebih dari n^{n-2} maka sistem tidak akan berfungsi.

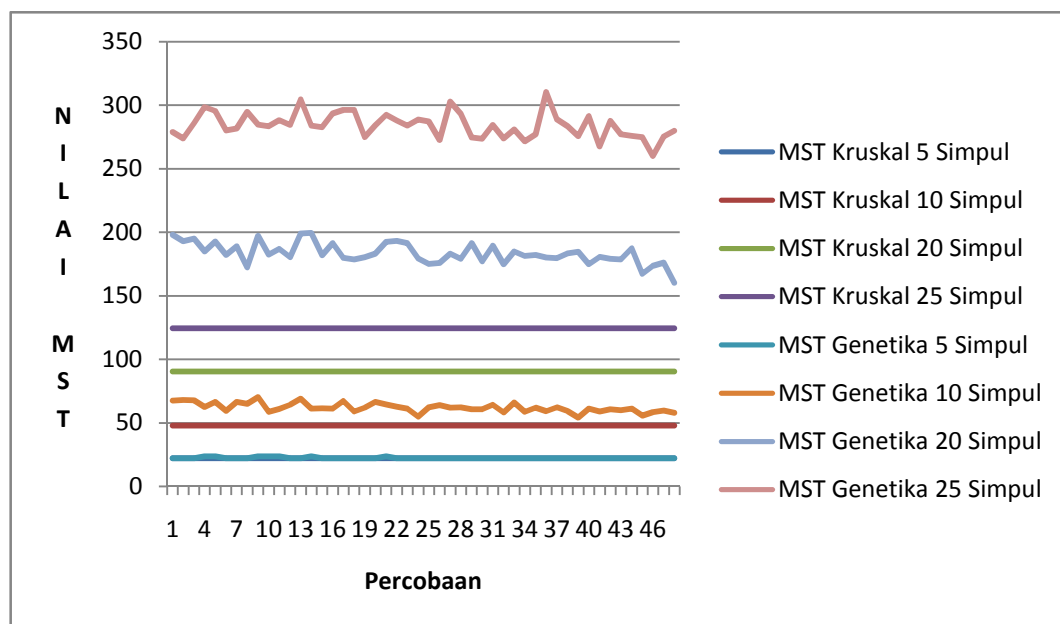
Dalam pengujian yang dilakukan saat ini, pengujian dilakukan dengan menggunakan jumlah data simpul yang berbeda, yaitu 5 simpul, 10 simpul, 20 simpul, dan 25 simpul dan parameter pululasi yaitu 2 x Jumlah simpul sampai 10 x jumlah simpul (tidak melebihi dari n^{n-2}) dan probabilitas yang digunakan adalah probalilitas *crossover* (*crossover* < 0.4), mutasi (mutasi < 0), serta seluruh populasi yang di-*generate* sebelumnya diikutkan kembali sebaagai generasi pada

iterasi selanjutnya (kelestarian = 1). Untuk Data Pengujian Lengkap, berupa data koordinat simpul, graf awal dan hasil pengujian dapat dilihat di Lampiran B.

Dari pengujian yang telah dilakukan, Rata-rata nilai MST yang dihasilkan oleh Algoritma Kruskal dan Algoritma Genetika adalah sebagai berikut :

Tabel 5.1 Rata-rata Hasil MST Kruskal dan Genetika

Simpul	MST Kruskal	MST Genetika
5	22.27	22.46
10	47.81	62.14
20	90.37	183.36
25	124.39	284.06



Gambar 5.13 Grafik Hasil Percobaan MST dengan Kruskal dan Genetika

Dari tabel 5.1 dan grafik 5.13, dapat dilihat hasil pengujian MST Algoritma Kruskal dan Algoritma Genetika. Algoritma Kruskal memberikan hasil nilai minimum yang tetap pada tiap percobaan yang dilakukan, hal ini dikarenakan kelemahan Algoritma Kruskal yang terjebak pada solusi lokal optimal, yaitu nilai MST yang dihasilkan akan selalu sama pada tiap percobaan.

Sedangkan algoritma Genetika lebih bergerak, hal ini disebabkan pada Algoritma Genetika, hasil MST yang dihasilkan juga dipengaruhi oleh parameter-parameter Algoritma Genetika yang diberikan. Walaupun kelemahan Algoritma Kruskal adalah terjebak pada solusi lokal optimal, namun nilai optimum yang dihasilkan oleh Algoritma Kruskal secara umum lebih baik dari nilai Minimum yang dihasilkan Algoritma Genetika.

Untuk rata-rata waktu proses MST Algoritma Kruskal dan Algoritma Genetika adalah pada tabel 5.2 :

Tabel 5.2 Rata-rata waktu proses MST Kruskal dan Genetika

Simpul	Waktu proses MST Kruskal	Waktu Proses MST Genetika
5	0.01373844	2.36437556
10	0.0172689	9.74124621
20	0.03916567	72.2387071
25	0.05063688	103.460496

Pada waktu proses MST dengan Algoritma Kruskal dan Algoritma Genetika, Waktu yang dibutuhkan Algoritma Kruskal juga lebih cepat dari Algoritma Genetika. Hal ini dikarenakan Algoritma Kruskal yang memiliki prinsip pencarian *Greedy* yaitu dengan acak langsung mencari nilai minimum dari graf, sedangkan Algoritma Genetika membutuhkan iterasi-iterasi yang panjang, sehingga membutuhkan waktu yang lebih lama dari Algoritma Kruskal

5.3 Analisis Kinerja Sistem

Seperti yang telah disebutkan pada subbab 3.4, analisis sistem menggunakan teknik rancangan percobaan, dimana hasil-hasil pengujian sebelumnya di analisis dengan bantuan aplikasi SPSS 22

5.3.1 Analisis Hasil MST Algoritma Kruskal dan Algoritma Genetika

5.3.1.1 Analisis Hasil Kruskal

Pada Algoritma kruskal, faktor yang mempengaruhi nilai *Minimum Spanning Tree* (MST) yang dihasilkan hanyalah jumlah simpul yang diberikan, oleh sebab itu digunakan rancangan percobaan random lengkap untuk mendapatkan tabel ANOVAny. Analisis ANOVA dilakukan untuk melihat dan membuktikan pengaruh dari perubahan-perubahan parameter seperti jumlah simpul yang diberikan, terhadap nilai MST yang dihasilkan.

Analisis ANOVA dilakukan pada data hasil percobaan (Lihat Lampiran B) dan menggunakan teknik rancangan percobaan random lengkap, hasil analisis ANOVA (*Analysis of Variance*) terhadap nilai *minimum spanning tree* yang dihasilkan oleh algoritma kruskal adalah :

Tabel 5.3 Tabel Analisis ANOVA Algoritma Kruskal

Descriptives						
MST						
	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean	
					Lower Bound	Upper Bound
Simpul 5	48	22.27	.000	.000	22.27	22.27
Simpul 10	48	47.81	.000	.000	47.81	47.81
Simpul 20	48	90.37	.000	.000	90.37	90.37
Simpul 25	48	124.39	.000	.000	124.39	124.39
Total	192	71.21	39.275	2.834	65.62	76.80

Tabel 5. 4 Tabel Deskripsi ANOVA

Descriptives		
MST		
	Minimum	Maximum
Simpul 5	22	22
Simpul 10	48	48
Simpul 20	90	90

Simpul 24	124	124
Total	22	124

Tabel 5. 5 Tabel Hasil Analisis ANOVA

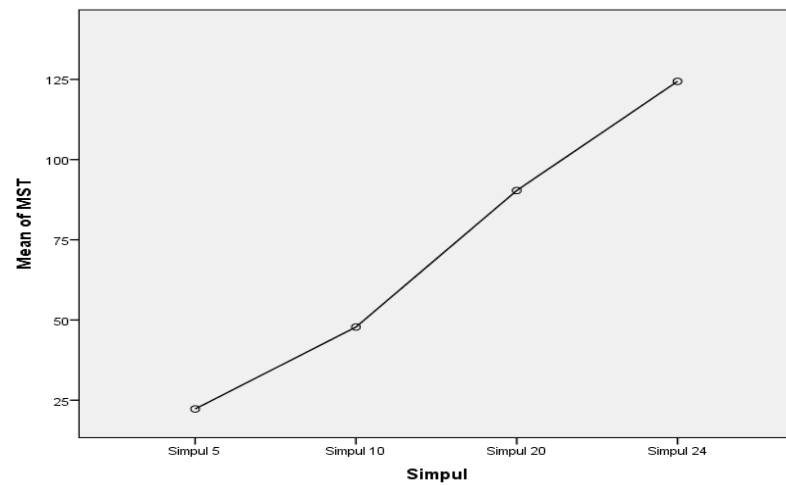
ANOVA					
MST					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	294619.277	3	98206.426	8.412E+31	.000
Within Groups	.000	188	.000		
Total	294619.277	191			

Analisa :

Dari Tabel 5.3, 5.4 dan 5.5 didapatkan kesimpulan nilai MST terkecil terdapat pada simpul 5, dengan nilai rata-rata MST 22.27, dan MST terbesar pada simpul 25 yaitu dengan nilai MST 124.39, dengan jumlah N adalah 48, dimana N adalah jumlah percobaan yang dilakukan. Standar Deviasi = 0, standar error = 0, yaitu tidak ada perubahan nilai pada MST dengan simpul yang sama dalam percobaan.

Untuk melihat hasil grafik analisa perbandingan nilai MST terhadap jumlah simpul dapat dilihat pada gambar 5.14. Dimana semakin banyak jumlah simpul, semakin naik nilai MST yang dihasilkan.

Means Plots



Gambar 5.14 Grafik hasil MST Kruskal

Analisa:

Dalam Rancangan Acak Lengkap tingkat signifikansi $\alpha = 0.05$, dan hipotesis yang diberikan untuk analisa efek perlakuan adalah :

H_0 = Perubahan nilai variabel (Simpul) pada percobaan, tidak mempengaruhi nilai MST yang dihasilkan

H_1 = Perubahan nilai variabel (Simpul) pada percobaan, mempengaruhi nilai MST yang dihasilkan

Dasar pengambilan keputusan yaitu tolak H_0 jika $\alpha > \text{sig.}$

Dan disimpulkan :

Oleh karena $\alpha = 0.05 > \text{sig.} = 0.000$ maka H_0 ditolak dan H_1 diterima. Dengan kata lain nilai MST pada Algoritma Kruskal sangat dipengaruhi oleh banyak nya simpul yang ada.

5.3.1.2 Analisis Hasil Genetika

Pada Algoritma genetika, ada beberapa faktor yang mempengaruhi nilai *Minimum Spanning Tree* (MST) yang dihasilkan, yaitu jumlah simpul, jumlah populasi, probabilitas *crossover*, dan nilai mutasi. Oleh sebab itu digunakan rancangan percobaan faktorial untuk mendapatkan tabel ANOVA nya.

Dengan menggunakan teknik rancangan percobaan faktorial, Hasil analisis ANOVA (*Analysis of Variance*) terhadap nilai *minimum spanning tree* yang dihasilkan oleh algoritma genetika adalah :

Tabel 5.6 Tabel Analisis ANOVA Algoritma Genetika

Between-Subjects Factors			
		Value Label	N
populasi	1	populasi 2x simpul	48
	2	populasi 3x simpul	48
	3	populasi 5x simpul	48
	4	populasi 10x simpul	48
crossover	1	crossover 0.1	64
	2	crossover 0.2	64
	3	crossover 0.3	64
mutasi	1	mutasi 0	48
	2	mutasi 0.1	48
	3	mutasi 0.2	48
	4	mutasi 0.3	48

Tabel 5. 7 Tabel Descriptive Statistics

Descriptive Statistics					
Dependent Variable: MST					
populasi	crossover	mutasi	Mean	Std. Deviation	N
populasi 2x simpul	crossover 0.1	mutasi 0	67.5900	.	1
		mutasi 0.1	67.9900	.	1
		mutasi 0.2	67.9300	.	1
		mutasi 0.3	62.5000	.	1
		Total	66.5025	2.67414	4

	crossover 0.2	mutasi 0	66.5300	.	1
		mutasi 0.1	59.6500	.	1
		mutaso 0.2	66.5200	.	1
		mutasi 0.3	65.0200	.	1
		Total	64.4300	3.26469	4
	crossover 0.3	mutasi 0	70.3700	.	1
		mutasi 0.1	58.9400	.	1
		mutaso 0.2	61.1000	.	1
		mutasi 0.3	64.4400	.	1
		Total	63.7125	4.98175	4
	Total	mutasi 0	68.1633	1.98316	3
		mutasi 0.1	62.1933	5.03260	3
		mutaso 0.2	65.1833	3.60586	3
		mutasi 0.3	63.9867	1.31975	3
		Total	64.8817	3.62659	12
populasi 3x simpul	crossover 0.1	mutasi 0	69.0000	.	1
		mutasi 0.1	61.3800	.	1
		mutaso 0.2	61.5600	.	1
		mutasi 0.3	61.3100	.	1
		Total	63.3125	3.79313	4
	crossover 0.2	mutasi 0	67.2900	.	1
		mutasi 0.1	59.0300	.	1
		mutaso 0.2	62.1100	.	1
		mutasi 0.3	66.5500	.	1
		Total	63.7450	3.88757	4
	crossover 0.3	mutasi 0	64.5400	.	1
		mutasi 0.1	62.8200	.	1
		mutaso 0.2	61.3900	.	1
		mutasi 0.3	55.1600	.	1
		Total	60.9775	4.08655	4
	Total	mutasi 0	66.9433	2.25012	3
		mutasi 0.1	61.0767	1.91312	3
		mutaso 0.2	61.6867	.37634	3
		mutasi 0.3	61.0067	5.70106	3
		Total	62.6783	3.76990	12
populasi 5x simpul	crossover 0.1	mutasi 0	62.3800	.	1
		mutasi 0.1	64.0300	.	1
		mutaso 0.2	62.1500	.	1
		mutasi 0.3	62.2300	.	1

	crossover 0.2	Total	62.6975	.89343	4
		mutasi 0	60.7300	.	1
		mutasi 0.1	60.8800	.	1
		mutaso 0.2	64.2400	.	1
		mutasi 0.3	58.9700	.	1
	crossover 0.3	Total	61.0300	2.45182	4
		mutasi 0	66.1200	.	1
		mutasi 0.1	58.8900	.	1
		mutaso 0.2	62.1600	.	1
		mutasi 0.3	59.3200	.	1
	Total	Total	61.6225	3.33089	4
		mutasi 0	63.0767	2.76171	3
		mutasi 0.1	61.2667	2.59172	3
		mutaso 0.2	62.8500	1.20379	3
		mutasi 0.3	59.9400	2.05151	3
populasi 10x simpul	crossover 0.1	Total	61.7833	2.32437	12
		mutasi 0	62.3800	.	1
		mutasi 0.1	59.2800	.	1
		mutaso 0.2	56.3100	.	1
		mutasi 0.3	61.2400	.	1
	crossover 0.2	Total	59.3025	3.56607	4
		mutasi 0	59.1900	.	1
		mutasi 0.1	60.7300	.	1
		mutaso 0.2	60.0400	.	1
		mutasi 0.3	61.3500	.	1
	crossover 0.3	Total	60.3275	.92809	4
		mutasi 0	55.8400	.	1
		mutasi 0.1	58.5900	.	1
		mutaso 0.2	59.9300	.	1
		mutasi 0.3	58.0200	.	1
	Total	Total	58.0950	1.70322	4
		mutasi 0	59.1367	3.27033	3
		mutasi 0.1	59.5333	1.09226	3
		mutaso 0.2	58.0933	3.27692	3
		mutasi 0.3	60.2033	1.89162	3
Total	crossover 0.1	Total	59.2417	2.32433	12
		mutasi 0	65.3375	3.46320	4
		mutasi 0.1	63.1700	3.75536	4
	crossover 0.2	mutaso 0.2	61.4875	5.58171	4

		mutasi 0.3	61.8200	.63953	4
		Total	62.9538	3.73845	16
	crossover 0.2	mutasi 0	63.4350	4.07337	4
		mutasi 0.1	60.0725	.88500	4
		mutaso 0.2	63.2275	2.78536	4
		mutasi 0.3	62.7975	3.72444	4
		Total	62.3831	3.12342	16
	crossover 0.3	mutasi 0	64.2175	6.10353	4
		mutasi 0.1	59.8100	2.01261	4
		mutaso 0.2	61.1450	.92529	4
		mutasi 0.3	59.2350	3.88077	4
		Total	61.1019	3.92525	16
	Total	mutasi 0	64.3300	4.31522	12
		mutasi 0.1	61.0175	2.77566	12
		mutaso 0.2	61.9533	3.42829	12
		mutasi 0.3	61.2842	3.23515	12
		Total	62.1462	3.61990	48

Analisa :

Rancangan Split plot di atas diuji dan dianalisa pada data simpul 10 yang dipengaruhi oleh 3 faktor yaitu populasi (populasi 2x simpul, populasi 3x simpul, populasi 5x simpul, populasi 10x simpul), *crossover* (0.1; 0.2; 0.3), dan mutasi (0; 0.1; 0.2; 0.3)

Dari tabel 5.6, dapat di analisa :

1. Pada populasi 2x simpul, hasil MST terbaik diperoleh pada *crossover* 0.3 dan mutasi 0.1 sebesar 58.9400
2. Pada populasi 3x simpul, hasil MST terbaik diperoleh pada *crossover* 0.3 dan mutasi 0.3 sebesar 55.1600
3. Pada populasi 5x simpul, hasil MST terbaik diperoleh pada *crossover* 0.3 dan mutasi 0.1 sebesar 58.8900

4. Pada populasi 10x simpul, hasil MST terbaik diperoleh pada *crossover* 0.3 dan mutasi 0 sebesar 55.8400
5. Nilai MST terbaik dihasilkan pada populasi 3x simpul, *crossover* 0.3 dan mutasi 0.3 sebesar 55.1600

Tests of Between-Subjects Effects						
Dependent Variable: MST						
Source		Type II Sum of Squares	df	Mean Square	F	Sig.
Intercept	Hypothesis	185383.507	1	185383.507	6792.351	.000
	Error	81.879	3	27.293 ^a		
populasi	Hypothesis	196.007	3	65.336	7.858	.007
	Error	74.831	9	8.315 ^b		
crossover	Hypothesis	28.782	2	14.391	1.983	.218
	Error	43.551	6	7.259 ^c		
mutasi	Hypothesis	81.879	3	27.293	4.428	.211
	Error	10.951	1.777	6.164 ^d		
populasi * crossover	Hypothesis	21.448	6	3.575	.380	.882
	Error	169.373	18	9.410 ^e		
populasi * mutasi	Hypothesis	74.831	9	8.315	.884	.557
	Error	169.373	18	9.410 ^e		
crossover * mutasi	Hypothesis	43.551	6	7.259	.771	.602
	Error	169.373	18	9.410 ^e		
populasi * crossover * mutasi	Hypothesis	169.373	18	9.410	.	.
	Error	.000	0	^f		
a. MS(mutasi)						
b. MS(populasi * mutasi)						
c. MS(crossover * mutasi)						
d. MS(populasi * mutasi) + MS(crossover * mutasi) - MS(populasi * crossover * mutasi)						
e. MS(populasi * crossover * mutasi)						
f. MS(Error)						

Analisa :

Dari analisis Rancangan Percobaan, nilai tingkat signifikansi ditetapkan, $\alpha = 0.05$.
dan variabel pengubah adalah Jumlah Populasi, *Crossover*, dan Mutasi.

Hipotesis :

H_0 = Perubahan variabel Populasi, *Crossover*, dan Mutasi tidak berpengaruh pada
nilai MST yang dihasilkan

H_1 = Perubahan variabel Populasi, *Crossover*, dan Mutasi berpengaruh pada nilai
MST yang dihasilkan

Dasar pengambilan keputusan adalah : Tolak H_0 jika $\alpha > \text{sig}$

Keputusan :

1. Pengaruh Populasi :

Karena $\alpha = 0.05 < \text{sig.} = 0.07$, maka H_0 ditolak dan H_1 diterima, atau dengan
kata lain bahwa perubahan populasi berpengaruh pada hasil MST genetika
yang dihasilkan.

2. Pengaruh *Crossover* :

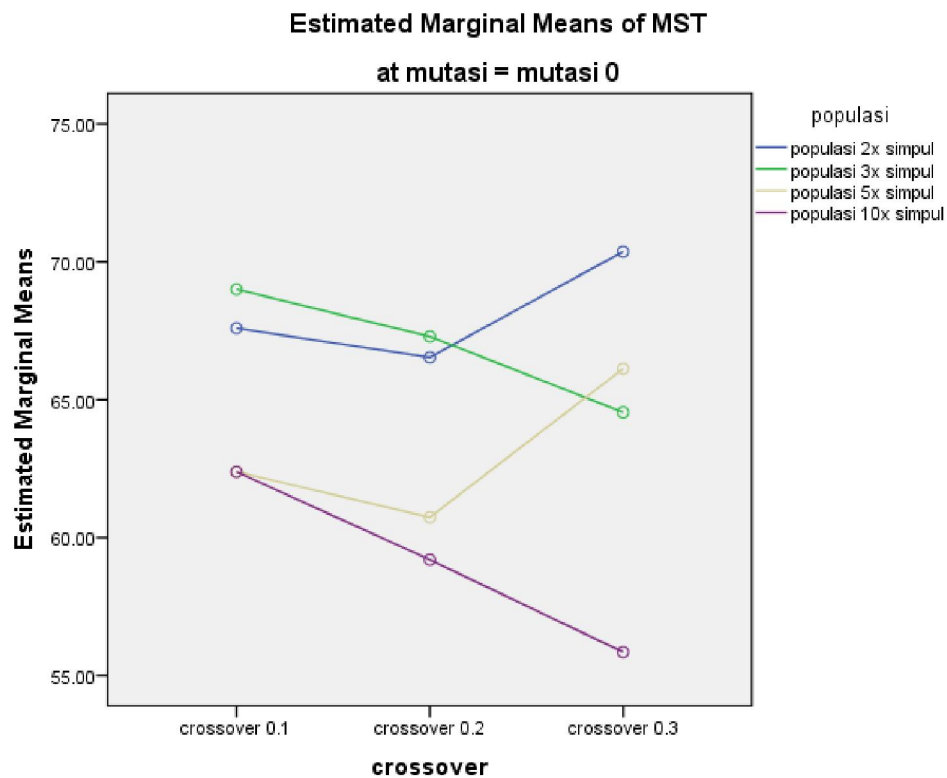
Karena $\alpha = 0.05 < \text{sig.} = 0.218$, maka H_0 ditolak dan H_1 diterima, atau
dengan kata lain bahwa perubahan nilai *crossover* berpengaruh pada hasil
MST genetika yang dihasilkan.

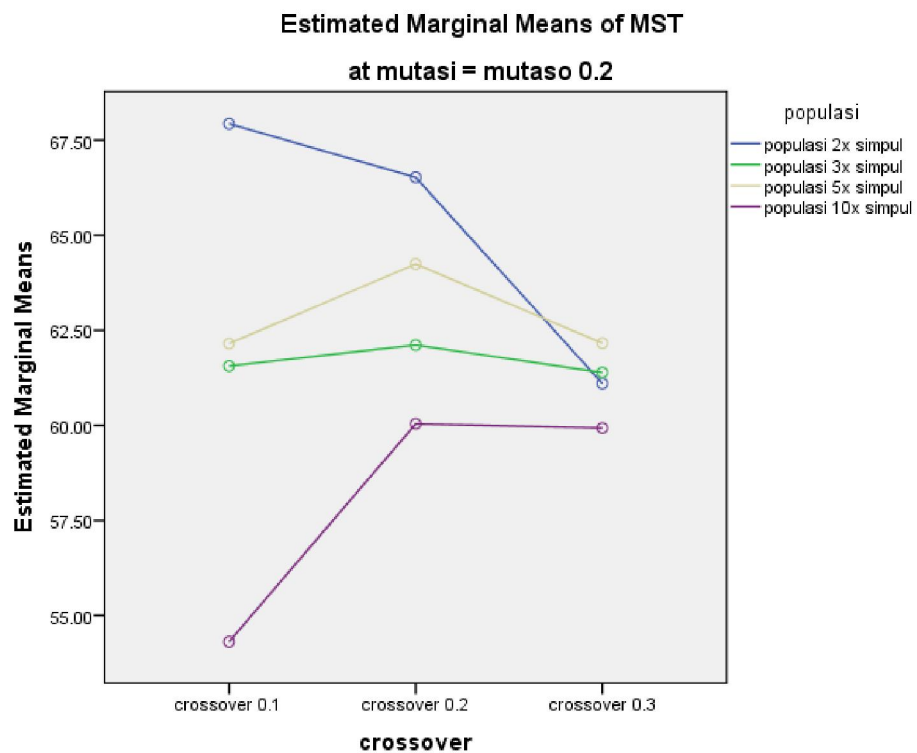
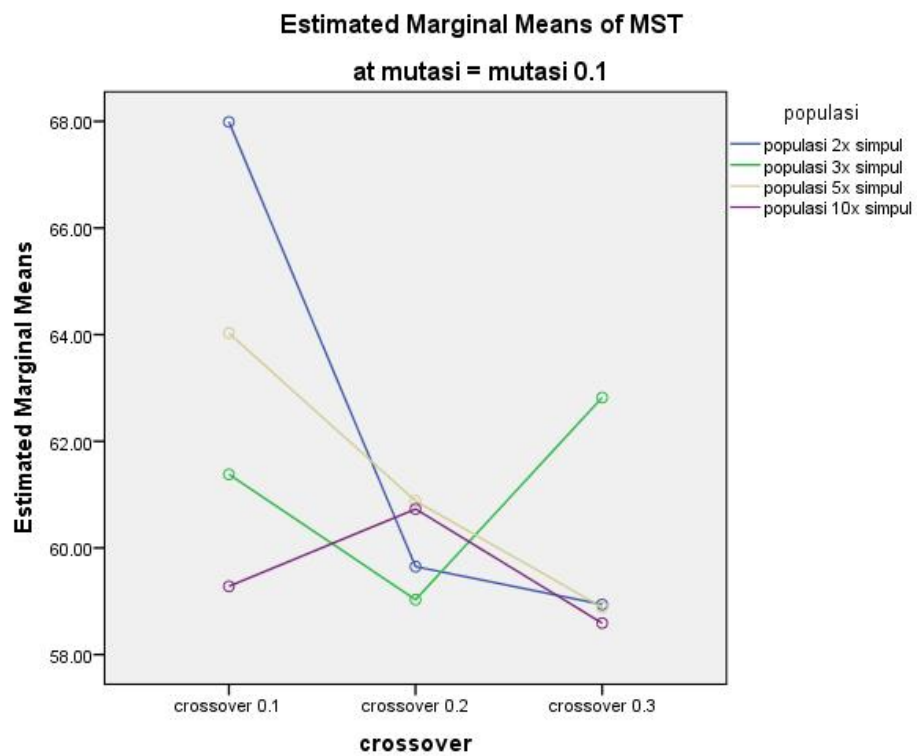
3. Pengaruh Mutasi :

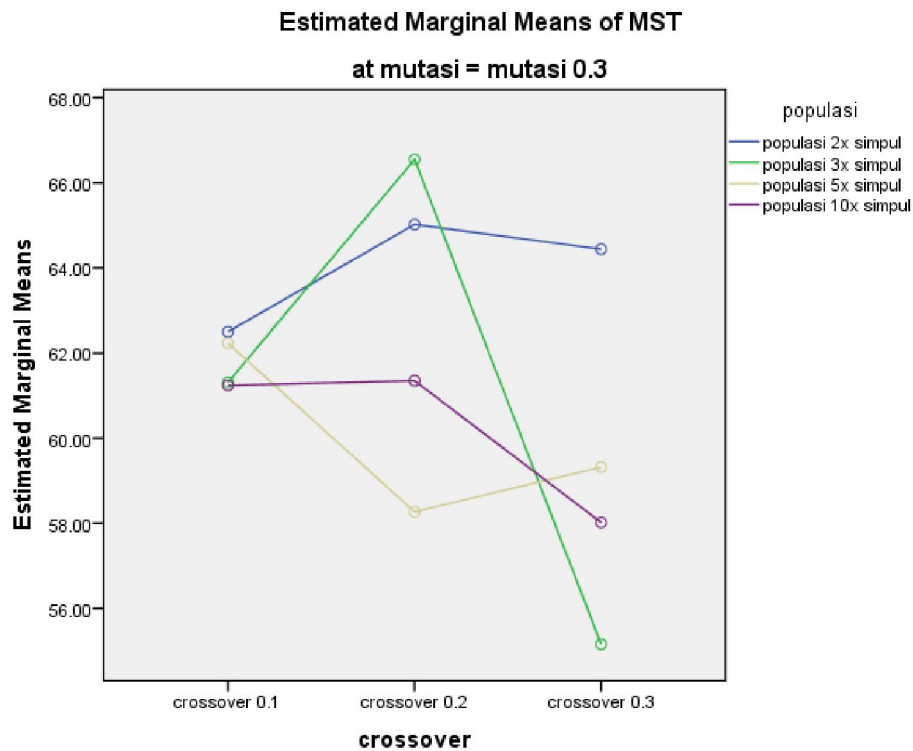
Karena $\alpha = 0.05 < \text{sig.} = 0.211$, maka H_0 ditolak dan H_1 diterima, atau
dengan kata lain bahwa perubahan nilai mutasi berpengaruh pada hasil
MST genetika yang dihasilkan.

Sedangkan $\text{sig.crossover} = 0.218 > \text{sig. mutasi} = 0.211$, disimpulkan perubahan dari *crossover* lebih besar pengaruhnya terhadap Algoritma Genetika dibandingkan perubahan nilai mutasi.

Profile Plot







Gambar 5. 15 Grafik perubahan parameter genetika terhadap nilai MST

Dari plot Gambar 5.15 diatas dapat dilihat, nilai MST minimum rata-rata dihasilkan pada *crossover* = 0.3.

5.3.2 Perbandingan Waktu Proses Algoritma Genetika dan Algoritma Kruskal

5.3.2.1 Waktu Proses Algoritma Kruskal

Pada Algoritma Kruskal, waktu proses algoritma hanya di pengaruhi oleh satu faktor yaitu Jumlah Simpul yang di-*input*-kan. Tabel analisis ANOVA lengkap dapat dilihat pada tabel 5.8 dan tabel 5.9

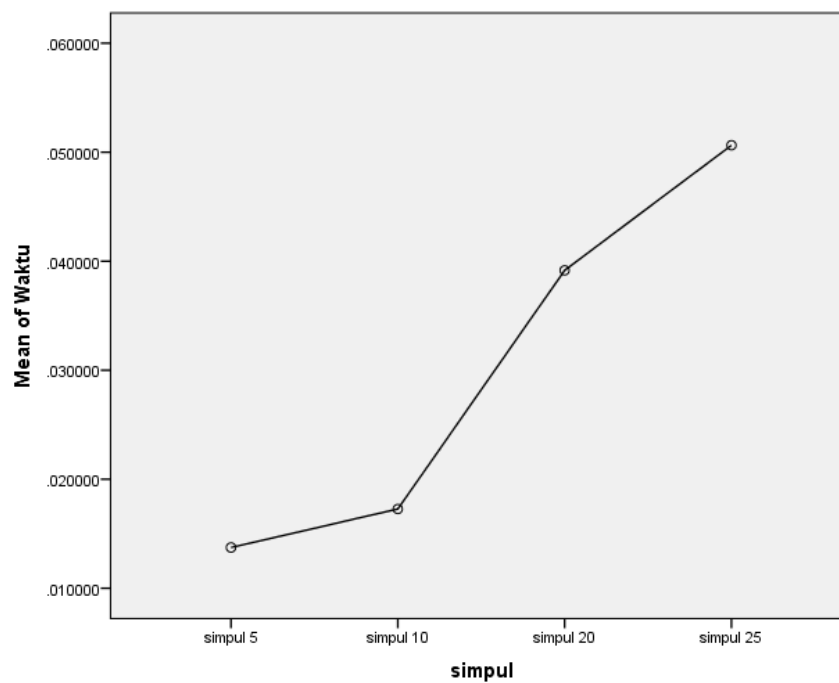
Tabel 5. 8 Tabel ANOVA kruskal terhadap waktu

Descriptives		
	Minimum	Maximum
simpul 5	.006105	.017445
simpul 10	.011005	.031313
simpul 20	.028491	.063876
simpul 25	.044964	.081283
Total	.006105	.081283

Tabel 5. 9 Tabel Hasil One-Way ANOVA

ANOVA					
Waktu					
	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	.045	3	.015	435.488	.000
Within Groups	.006	188	.000		
Total	.051	191			

Means Plots



Gambar 5.16 Grafik perbandingan jumlah simpul dan perubahan Waktu

Dari Gambar 5.16 diatas disimpulkan, jumlah waktu proses yang dibutuhkan oleh Algoritma Kruskal dalam menghitung nilai MST sangat berpengaruh pada jumlah simpul. Semakin besar jumlah simpul, maka akan semakin lama waktu yang dibutuhkan oleh Algoritma Kruskal memproses nilai MST.

5.3.2.2 Waktu Proses Algoritma Genetika

Pada Algoritma genetika, waktu proses algoritma dipengaruhi oleh Jumlah Simpul dan jumlah populasi yang di-*input*-kan. Tabel analisis ANOVA Pengaruh jumlah simpul dan jumlah Populasi terhadap waktu proses Algoritma Genetika lengkap dapat dilihat pada tabel 5.10 sampai tabel 5.13

Tabel 5. 10 Tabel ANOVA Algoritma Kruskal

Between-Subjects Factors			
		Value Label	N
simpul	1	Simpul 5	48
	2	Simpul 10	48
	3	Simpul 20	48
	4	Simpul 25	48
populasi	1	populasi 2x simpul	48
	2	populasi 3x simpul	48
	3	populasi 5x simpul	48
	4	populasi 10x simpul	48

Tabel 5. 11 Tabel tes dari Subjek-Efek ANOVA

Tests of Between-Subjects Effects					
Dependent Variable: waktu					
Source	Type III Sum of Squares	df	Mean Square	F	Sig.
Corrected Model	537929.761 ^a	6	89654.960	71.699	.000
Intercept	423247.827	1	423247.827	338.482	.000
simpul	345855.558	3	115285.186	92.197	.000
populasi	192074.203	3	64024.734	51.202	.000
Error	231329.274	185	1250.429		

Total	1192506.862	192			
Corrected Total	769259.035	191			

a. R Squared = .699 (Adjusted R Squared = .690)

Post Hoc Tests

simpul

Homogeneous Subsets

Tabel 5. 12 Tabel ANOVA Perubahan Simpul

waktu				
Duncan ^{a,b}				
simpul	N	Subset		
		1	2	3
Simpul 5	48	2.36437556		
Simpul 10	48	9.74124621		
Simpul 20	48		72.23870710	
Simpul 25	48			103.4604960 2
Sig.		.308	1.000	1.000

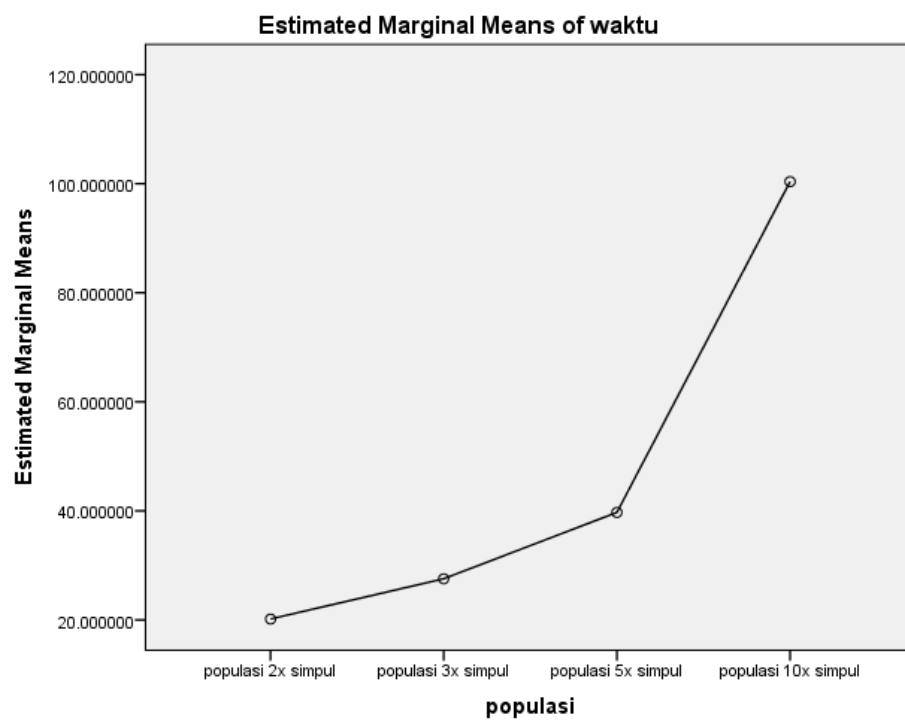
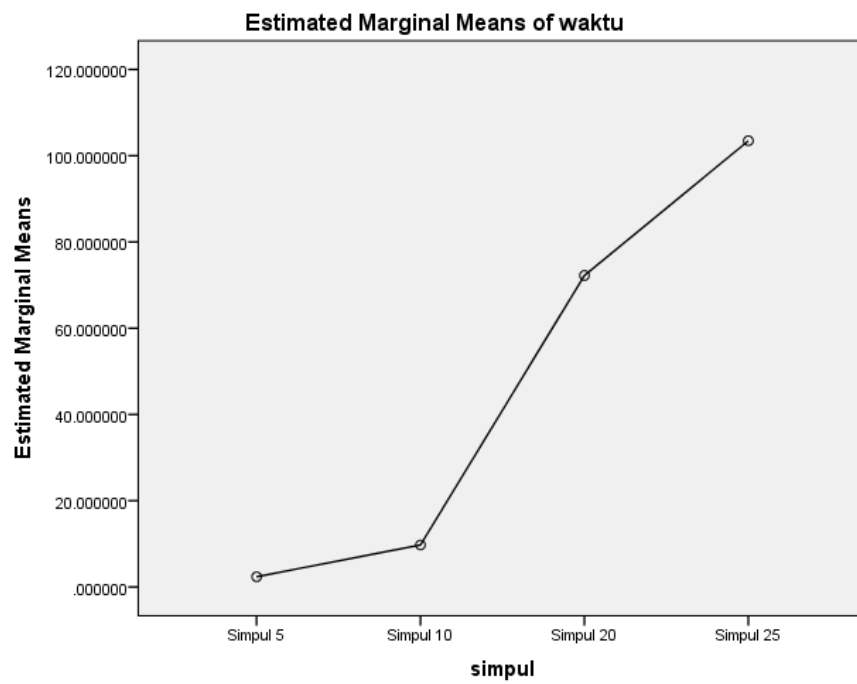
populasi

Homogeneous Subsets

Tabel 5. 13 Tabel ANOVA Perubahan Populasi

waktu				
Duncan ^{a,b}				
populasi	N	Subset		
		1	2	3
populasi 2x simpul	48	20.16108960		
populasi 3x simpul	48	27.54826119	27.54826119	
populasi 5x simpul	48		39.71242929	
populasi 10x simpul	48			100.38304481
Sig.		.307	.094	1.000

Profile Plots



Gambar 5. 17 Grafik perubahan waktu Algoritma Genetika

Dari plot waktu Gambar 5.17 diatas, disimpulkan jumlah waktu proses yang dibutuhkan oleh Algoritma Genetika dalam menghitung nilai MST sangat berpengaruh pada jumlah simpul dan populasi. Semakin besar jumlah simpul dan populasi, maka akan semakin lama waktu yang dibutuhkan oleh Algoritma genetika memproses nilai MST.

BAB VI

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil penelitian, analisis dan pembahasan yang telah dilakukan, maka dapat disimpulkan :

1. Penelitian ini menghasilkan aplikasi penyelesaian masalah *Minimum Spanning Tree* (MST) menggunakan Algoritma Kruskal dan Algoritma Genetika.
2. Dari hasil pengujian untuk simpul < 25 , didapatkan beberapa kesimpulan yaitu :
 - a. Algoritma Kruskal menghasilkan nilai jarak minimum untuk penyelesaian permasalahan *Minimum Spanning Tree* lebih baik dari Algoritma genetika, dimana nilai MST yang dihasilkan yaitu untuk simpul = 5, nilai MST yang dihasilkan kruskal adalah 22.27, sedangkan MST genetika = 22.46. untuk simpul = 10, nilai MST kruskal = 47.81, sedangkan MST genetika = 62.14, untuk simpul = 20, nilai MST kruskal = 90.37, sedangkan MST genetika = 183.36 untuk simpul = 25, nilai MST kruskal = 124.39, sedangkan MST genetika = 284.06
 - b. Waktu proses yang dibutuhkan oleh Algoritma Kruskal dalam menyelesaikan permasalahan *Minimum Spanning Tree* (MST) jauh lebih cepat dibandingkan Algoritma Genetika, dimana rata-rata waktu proses algoritma Kruskal adalah 0.030 detik, dan Algoritma Genetika adalah 46.95 detik.

3. Dari tabel ANOVA (*Analisis of Varians*) didapatkan :
- a. Nilai MST Algoritma Kruskal hanya dipengaruhi oleh jumlah simpul, semakin banyak jumlah simpul yang di-*input*-kan, maka akan semakin besar nilai MST yang dihasilkan
 - b. Pada Algoritma Genetika, Jumlah Populasi, Probabilitas *Crossover* dan Probabilitas Mutasi sangat mempengaruhi nilai MST
 - c. Algoritma Genetika akan menghasilkan nilai minimum yang cukup baik pada saat parameter peluang *crossover* kecil, yaitu 0.1 – 0.3 dari populasi di *crossover*, dan mutasi = 0 serta kelestarian = 1 , dimana seluruh populasi awal yg telah di *generate* dijadikan individu untuk iterasi selanjutnya.
 - d. Waktu proses Algoritma Kruskal dipengaruhi hanya oleh jumlah simpul, sedangkan pada Algoritma Genetika, waktu proses dipengaruhi oleh jumlah simpul dan jumlah populasi yang di-*input*-kan.

5.2. Saran

Berdasarkan hasil penelitian, pengujian dan pembahasan yang telah dipaparkan, maka penulis menyarankan untuk mengembangkan penelitian dimasa yang akan datang sebagai berikut :

1. Permasalahan graf berkembang menjadi jenis graf lain, misalnya direct graf, graf non-simetris, dsb
2. Penelitian dapat dikembangkan dan diperluas untuk pencarian jarak minimum dengan menerapkannya langsung pada studi kasus dengan jarak yang sebenarnya
3. Penelitian dapat dikembangkan dengan membandingkan Algoritma Kruskal dan/atau Algoritma Genetika dengan algoritma-algoritma lainnya, sehingga didapatkan algoritma yang paling baik dalam penyelesaian permasalahan nilai minimum pada suatu graf.

DAFTAR PUSTAKA

- Arikunto, S. (2010). *Prosedur Penelitian*. Rineka Cipta.
- Astuti, L. Y. (2013). *Perbandingan Kinerja Algoritma Genetika Pada Penyelesaian Masalah TSP (Traveling Salesman Problem) dan MST (Minimum Spanning Tree)*. Bengkulu.
- Ayu. (2012). *Pohon (Tree)*,
[http://ayu_ws.staff.gunadarma.ac.id/Downloads/files/33383/05+Pohon+\(Tree\).pdf](http://ayu_ws.staff.gunadarma.ac.id/Downloads/files/33383/05+Pohon+(Tree).pdf).
- Bilqis Amaliah, C. F. (2009). Optimasi Irigasi Sawah dengan Menggunakan Algoritma Genetika. *Digital Information and System Conference*.
- Efendi, R. (2003). *Penerapan Algoritma Semut Untuk Pemecahan Masalah Spanning Tree Pada Kasus Pemasangan Karingan Kabel Telepon*. Yogyakarta.
- Febriansyah. (2013). Metode Pengembangan Sistem Informasi.
<http://febriansyah.blog.widyatama.ac.id/2013/10/16/rekayasa-perangkat-lunak-metode-pengembangan-sistem-informasi/>.
- Kusumadewi, S. (2003). *Artificial Intelligence Teknik dan Aplikasinya*. Yogyakarta: Graha Ilmu.
- Kusumadewi, S., & Purnomo, H. (2005). *Penyelesaian Optimasi dengan Menggunakan teknik-teknik Heuristik*. Yogyakarta: Graha Ilmu.
- Lubis, I. H. (2011). *Studi Perbandingan Algoritma Prim Algoritma Kruskal Dan Algoritma Sollin Dalam Menentukan Pohon Merentang Maksimum*. Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Sumatera Utara.
- Munir, R. (2006). *Diktat Kuliah IF2153 Matematika Diskrit Edisi Keempat*. Bandung: Departemen Teknik Informatika, Institut Teknologi Bandung.
- Munir, R. (2012). *Matematika Dskrit Revisi Kelima*. Bandung: Informatika.
- Pramesti, G. (2005). *Mahir Menggunakan SPSS 13.0 dalam Ran Percobaan*. Jakarta: Elex Media Komputindo.
- Pratisto, A. (2005). *Cara Mudah Mengatasi Masalah Statistik dan Ran Percobaan dengan SPSS 12*. Jakarta: Elex Media Komputindo.
- Raupong, A. (2011). *Bahan Ajar Mata Kuliah Perancangan Percobaan. Program Studi Statistika*. Jurusan Matematika Fakultas Matematika & Ilmu pengetahuan Alam Universitas hasanuddin.

Sholihah, U. (2013). Teori Graph. <http://blog.iain-tulungagung.ac.id/ummusholihah/wp-content/uploads/sites/20/2013/11/teori-graph-ummu.docx>.

Simorangkir, S. H. (2010). *Studi Strategi Penggunaan Algoritma Greedy Untuk membangun Minimum Spanning Tree Pada Graf Berbobot (Weighted Graph)*.

Zhao, J. L. (2010). Application Development Approaches.

LAMPIRAN

Lampiran A. Hasil Implementasi Prosedur

1. Algoritma Kruskal

Pada algoritma kruskal, jarak *edge* terkecil dipilih secara acak, kemudian simpan kedalam tree, proses ini diulang sampai semua simpul terhubung dan membentuk *tree* minimum, dan tidak ada nya pelanggaran (membentuk sirkuit)

Coding mencari *edge* dengan jarak terpendek :

```
shortestEdge = biggestEdge
For row = 1 To numberOfNode
  For col = 1 To numberOfNode
    If matrix(row, col) < shortestEdge And
matrix(row, col) > 0 Then
      shortestEdge = matrix(row, col)
      startNode = row
      endNode = col
    End If
  Next
Next

If shortestEdge = biggestEdge Then Exit Sub
```

Coding Untuk Mengecek palanggaran (terjadinya Surkuit atau tidak)

```
Do While startNode <> endNode

  For pStart = 1 To numberOfNode
    If matrix(startNode, pStart) < 0 And pStart <>
pEnd Then
      stackPointer = stackPointer + 1
      stackStart(stackPointer) = startNode
      stackEnd(stackPointer) = pStart
    End If
  Next

  If stackPointer > 0 Then
    startNode = stackEnd(stackPointer)
    pEnd = stackStart(stackPointer)
    stackPointer = stackPointer - 1
  Else
    isCyclic = False
    Exit Function
  End If
Loop
isCyclic = True
```

2. Algoritma Genetika

Pada Algoritma genetika, proses yang dilakukan adalah Bangkitkan populasi awal dengan proses *Frufter Number* yang kemudian dapat di evaluasi (*decoding tree* untuk mendapatkan bobot yang digunakan untuk menghitung Nilai *fitness*, Probablilitas dan komulatif) kemudian dapat di *crossover*, mutasi dan kromosom yg telah di *generate* dilestarikan untuk dijadikan individu awal pada proses iterasi selanjutnya

Coding Frufer Number adalah sebagai berikut :

```
i = 1
Do While i < bil - 1
    hasil = Int(Rnd * bil) + 1
    K_Tree(i) = hasil
    i = i + 1
Loop
kal = ""
For i = 1 To bil - 2
    kal = kal & K_Tree(i) & " "
    Elkrom(i) = K_Tree(i)
Next i
Acak_Prufer_Number = kal
```

Hitung Bobot Individu :

```
Bobot_Tree(kromo) = 0
For i = 1 To JumTitik - 1
    Edge_Tree(kromo, i) = ki(i) & "-" & ka(i)
    EdgeKi(kromo, i) = ki(i)
    EdgeKa(kromo, i) = ka(i)
    Bobot_Tree(kromo) = Bobot_Tree(kromo) + m(ki(i), ka(i))
Next i
```

Selanjutnya Coding untuk Evaluasi kromosom:

```
For i = 1 To JumPopulasi
    If BobotKromo_Tree(i) > 0 Then
        FitnessperKromo_Tree(i) = 1 / BobotKromo_Tree(i)
    End If
Next i

TotalFitness_Tree = 0
For i = 1 To JumPopulasi
    TotalFitness_Tree = TotalFitness_Tree +
    FitnessperKromo_Tree(i)
```

```

Next i

For i = 1 To JumPopulasi
    If BobotKromo_Tree(i) > 0 Then
        ProbabilitasperKromo_Tree(i) =
        FitnessperKromo_Tree(i) / TotalFitness_Tree
    End If
Next i

For i = 1 To JumPopulasi
    HsProb = 0
    For j = 1 To i
        HsProb = HsProb + ProbabilitasperKromo_Tree(j)
    Next j
    KumulatifperKromo_Tree(i) = HsProb
Next i

```

Kemudian *Crossover* :

```

'Acak bil 0-1 (Rk)

For i = 1 To JumPopulasi
    hasil = Rnd * 1
    Rk_tree(i) = hasil
Next i

'Memilih Kromosom yang akan di_crossover
If Generasi > 1 Then
    n = 1
    For i = 1 To JumPopulasi
        If (Rk_tree(i) < ProbCrossover) And (i <>
        NoKromdgFitnessTertinggi) Then
            JumKromTerpilih = n
            n = n + 1
        End If
    Next i
Else
    n = 1
    For i = 1 To JumPopulasi
        If Rk_tree(i) < ProbCrossover Then
            JumKromTerpilih = n
            n = n + 1
        End If
    Next i
End If

If JumKromTerpilih > 1 Then
    ReDim NoSelectedKromToCross(JumKromTerpilih)
    If Generasi > 1 Then
        n = 1
        For i = 1 To JumPopulasi
            If Rk_tree(i) < ProbCrossover And (i <>
            NoKromdgFitnessTertinggi) Then
                NoSelectedKromToCross(n) = i      'Mengamankan
                kromosom yang tidak dipilih
                n = n + 1
            End If
        Next i
    End If
End If

```

```

        End If
    Next i
Else
    n = 1
    For i = 1 To JumPopulasi
        If Rk_tree(i) < ProbCrossover Then
            NoSelectedKromToCross(n) = i      'Mengamankan
kromosom yang tidak dipilih
            n = n + 1
        End If
    Next i
End If
tes = JumKromTerpilih Mod 2
ganjil = False
If tes > 0 Then
    ganjil = True
End If

If ganjil = True Then
    Dim Temp() As Integer
    ReDim Temp(JumKromTerpilih)
    For i = 1 To JumKromTerpilih - 1
        Temp(i) = NoSelectedKromToCross(i)
    Next
    JumKromTerpilih = JumKromTerpilih - 1
    For i = 1 To JumKromTerpilih
        NoSelectedKromToCross(i) = Temp(i)
    Next
End If

'Mencari Posisi Gen yang akan di-crossover

i = 1
Do While i < JumKromTerpilih + 1
    hs1 = Int(Rnd * (JumSimpul - 4)) + 1
    CutPoint(i) = hs1
    CutPoint(i + 1) = hs1
    i = i + 2
Loop

Dim tetap1 As String
Dim RightMove1 As String
Dim tetap As String
Dim RightMove As String
Dim Jومتetap As Integer
ReDim O_Tree(JumKromTerpilih)
Dim k As Integer
Dim TtkPotong As Integer
Dim JumTtpUrut As Integer

j = 1
Do While j < JumKromTerpilih + 1
    tetap1 = ""
    JumTtpUrut = CutPoint(j)
    TtkPotong = CutPoint(j)
    Dim Temp1() As Integer

```



```

ReDim Temp1(JumTitik - 2)
Dim Temp2() As Integer
ReDim Temp2(JumTitik - 2)

For i = TtkPotong + 1 To JumSimpul - 2
    Temp1(i) = Kromosom(NoSelectedKromToCross(j), i)
Next i

For i = TtkPotong + 1 To JumSimpul - 2
    Temp2(i) = Kromosom(NoSelectedKromToCross(j +
1), i)
Next i

For i = TtkPotong + 1 To JumSimpul - 2
    Kromosom(NoSelectedKromToCross(j), i) = Temp2(i)
Next i

For i = TtkPotong + 1 To JumSimpul - 2
    Kromosom(NoSelectedKromToCross(j + 1), i) =
Temp1(i)
Next i

j = j + 2
Loop

```

Coding Mutasi :

```

'Memilih Kromosom yang akan di_mutasi
If Generasi > 1 Then
    n = 1
    For i = 1 To JumPopulasi
        If Rm_tree(i) < ProbMutasi And (i <>
NoKromdgFitnessTertinggi) Then
            JumKromToMutasi = n
            n = n + 1
        End If
    Next i
Else
    n = 1
    For i = 1 To JumPopulasi
        If Rm_tree(i) < ProbMutasi Then
            JumKromToMutasi = n
            n = n + 1
        End If
    Next i
End If
If JumKromToMutasi > 0 Then
    ReDim NoSelectedKromToMutation(JumKromToMutasi)
    If Generasi > 1 Then
        n = 1
        For i = 1 To JumPopulasi
            If Rm_tree(i) < ProbMutasi And (i <>
NoKromdgFitnessTertinggi) Then
                NoSelectedKromToMutation(n) = i
'Mengamankan kromosom yang tidak dipilih
                n = n + 1
            End If
        Next i
    End If

```

```

Else
    n = 1
    For i = 1 To JumPopulasi
        If Rm_tree(i) < ProbMutasi Then
            NoSelectedKromToMutation(n) = i
'Mengamankan kromosom yang tidak dipilih
            n = n + 1
        End If
    Next i

End If
'Mencari posisi gen untuk mutasi setiap kromosom
ReDim PosMut(JumKromToMutasi, 2)
i = 1
Do While i < JumKromToMutasi + 1
    hs1 = Int(Rnd * (PanjangKromosom - 2)) + 1
    Pos(1) = hs1
    PosMut(i, 1) = Pos(1)
    hs2 = Int(Rnd * (PanjangKromosom - 1)) + 1
    If hs2 <= hs1 Then
        hs3 = Int(Rnd * (PanjangKromosom - hs1)) + 1
        hs2 = hs1 + hs3
    End If
    Pos(2) = hs2
    PosMut(i, 2) = Pos(2)
    i = i + 1
Loop

Dim Temp1 As Integer
Dim Temp2 As Integer
j = 1
Do While j < JumKromToMutasi + 1
    Temp1 = Kromosom(NoSelectedKromToMutation(j),
PosMut(j, 1))
    Temp2 = Kromosom(NoSelectedKromToMutation(j),
PosMut(j, 2))
    Kromosom(NoSelectedKromToMutation(j), PosMut(j, 1))
= Temp2
    Kromosom(NoSelectedKromToMutation(j), PosMut(j, 2))
= Temp1
    j = j + 1
Loop

```

Mencatat kelestarian kromoso yang akan dijadikan individu pada iterasi

selanjutnya :

```

FitnessMaksimum = FitnessperKromo_Tree(1)
NKrom = 1
For i = 2 To JumPopulasi
    If FitnessperKromo_Tree(i) > FitnessMaksimum Then
        FitnessMaksimum = FitnessperKromo_Tree(i)
        NKrom = i
    End If
Next i

```

```

'Memilih Kromosom yang akan diGanti
n = 1
For i = 1 To JumPopulasi
    If Rb_Tree(i) < ProbKelestarian And i <> NKrom Then
        JumKromToChange = n
        n = n + 1
    End If
Next i
ReDim NoSelectedKromToChange(JumPopulasi)
n = 1
For i = 1 To JumPopulasi
    If Rb_Tree(i) < ProbKelestarian And i <> NKrom Then
        NoSelectedKromToChange(n) = i      'Mengamankan kromosom
yang tidak dipilih
        n = n + 1
    End If
Next i

'Mengganti Kromosom yang Nilai Rb-nya < Prob kelestarian
If JumKromToChange > 0 Then

    Dim Temp() As Integer
    ReDim Temp(JumKromToChange)

    For i = 1 To JumKromToChange
        For j = 1 To JumSimpul - 2
            Kromosom(NoSelectedKromToChange(i), j) =
KromosomAwal(NoSelectedKromToChange(i), j)
        Next j
    Next i
End If

For i = 1 To JumPopulasi
    Modifikasi_Derajat i, JumSimpul
Next i

For i = 1 To JumPopulasi
    For j = 1 To JumSimpul - 2
        KromosomAkhirGenerasi(i, j) = Kromosom(i, j)
    Next j
Next i

FitnessMaksimum = FitnessperKromo_Tree(1)
NoKrom = 1
For i = 2 To JumPopulasi
    If FitnessperKromo_Tree(i) > FitnessMaksimum Then
        FitnessMaksimum = FitnessperKromo_Tree(i)
        NoKrom = i
    End If
Next i
NoKromdgFitnessTertinggi = NoKrom
BobotKromosomDgFitnessTertinggi = BobotKromo_Tree(NoKrom)
ReDim EdgeOptKi(JumSimpul - 1)
ReDim EdgeOptKa(JumSimpul - 1)

For i = 1 To JumSimpul - 1

```

```

        EdgeOptKi(i) = EdgeKi(NoKromdgFitnessTertinggi, i)
        EdgeOptKa(i) = EdgeKa(NoKromdgFitnessTertinggi, i)
    Next i

```

Terakhir adalah menentukan nilai optimal :

```

For i = 1 To 1000
    Crossover JumTitik, JumlahPopulasi, ProbCrossover
    Mutasi JumlahPopulasi, JumTitik, ProbMutasi
    Mencatat_PopulasiAkhir_Generasi ProbKelestarian,
    JumlahPopulasi, JumTitik, Generasi
    Generasi = Generasi + 1
Next i
Generasi = 0
ReDim MatrixGenetika(JumTitik, JumTitik)

For i = 1 To JumTitik - 1
    MatrixGenetika(EdgeOptKi(i), EdgeOptKa(i)) =
m(EdgeOptKi(i), EdgeOptKa(i))
    MatrixGenetika(EdgeOptKa(i), EdgeOptKi(i)) =
m(EdgeOptKi(i), EdgeOptKa(i))
Next i
BobotKromosomOptimal = BobotKromosomDgFitnessTertinggi

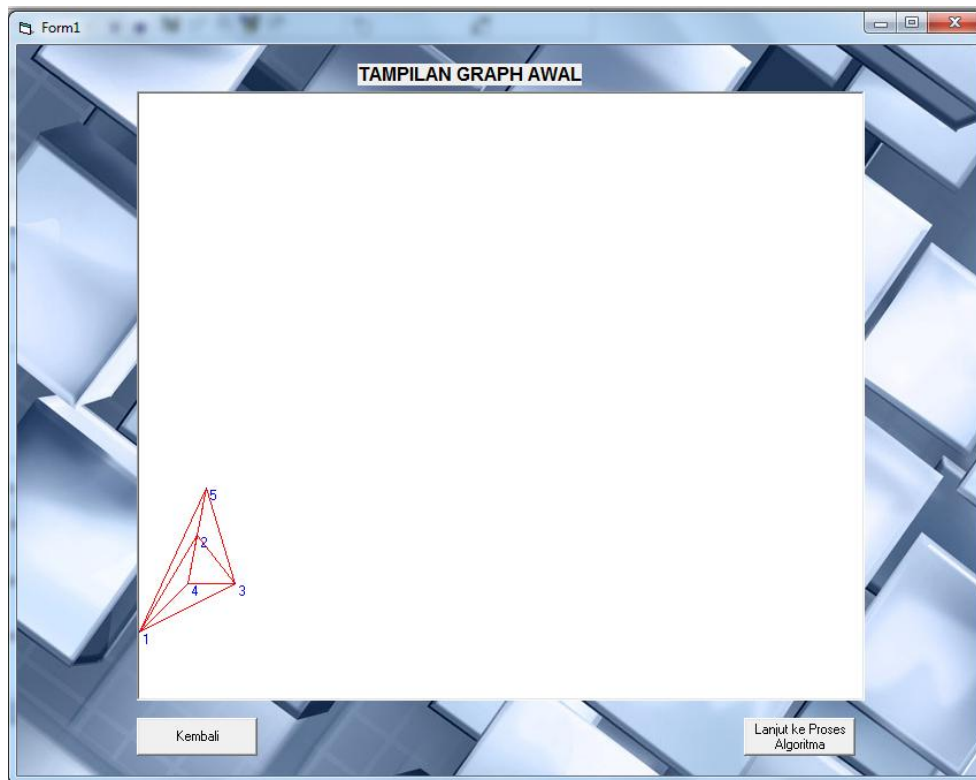
```

Lampiran B. Data Simpul, dan Grafik Awal nya

1. Data 5 Simpul

Simpul	X	Y
1	0	0
2	6	10
3	10	5
4	5	5
5	7	15

Graf Awal :



Jumlah Simpul : 5
 Populasi : 2x jumlah simpul s/d 10 x Jumlah Simpul
 Crossover : 0.1 - 0.3
 Mutasi : 0 - 0.3
 Kelestarian : 1

Percobaan	Simpul	Populasi	Crossover	Mutasi	Kelestarian	MST Kruskal	Waktu	MST Genetika	Waktu
1	5	10	0.1	0	1	22.27	0.014459	22.27	0.369836
2	5	10	0.1	0.1	1	22.27	0.01151	22.27	0.5015
3	5	10	0.1	0.2	1	22.27	0.006738	22.27	0.563792
4	5	10	0.1	0.3	1	22.27	0.015775	23.57	0.59434
5	5	10	0.2	0	1	22.27	0.006175	23.57	0.430043
6	5	10	0.2	0.1	1	22.27	0.015318	22.27	0.6074
7	5	10	0.2	0.2	1	22.27	0.007358	22.27	0.724504
8	5	10	0.2	0.3	1	22.27	0.016675	22.27	0.672898
9	5	10	0.3	0	1	22.27	0.015379	23.67	0.491057
10	5	10	0.3	0.1	1	22.27	0.014928	23.57	0.650854
11	5	10	0.3	0.2	1	22.27	0.011409	23.67	0.716337
12	5	10	0.3	0.3	1	22.27	0.015164	22.27	0.752915
13	5	15	0.1	0	1	22.27	0.016173	22.27	0.858024
14	5	15	0.1	0.1	1	22.27	0.006105	23.57	1.28022
15	5	15	0.1	0.2	1	22.27	0.015458	22.27	1.323277
16	5	15	0.1	0.3	1	22.27	0.015075	22.27	1.370462
17	5	15	0.2	0	1	22.27	0.015085	22.27	1.034422

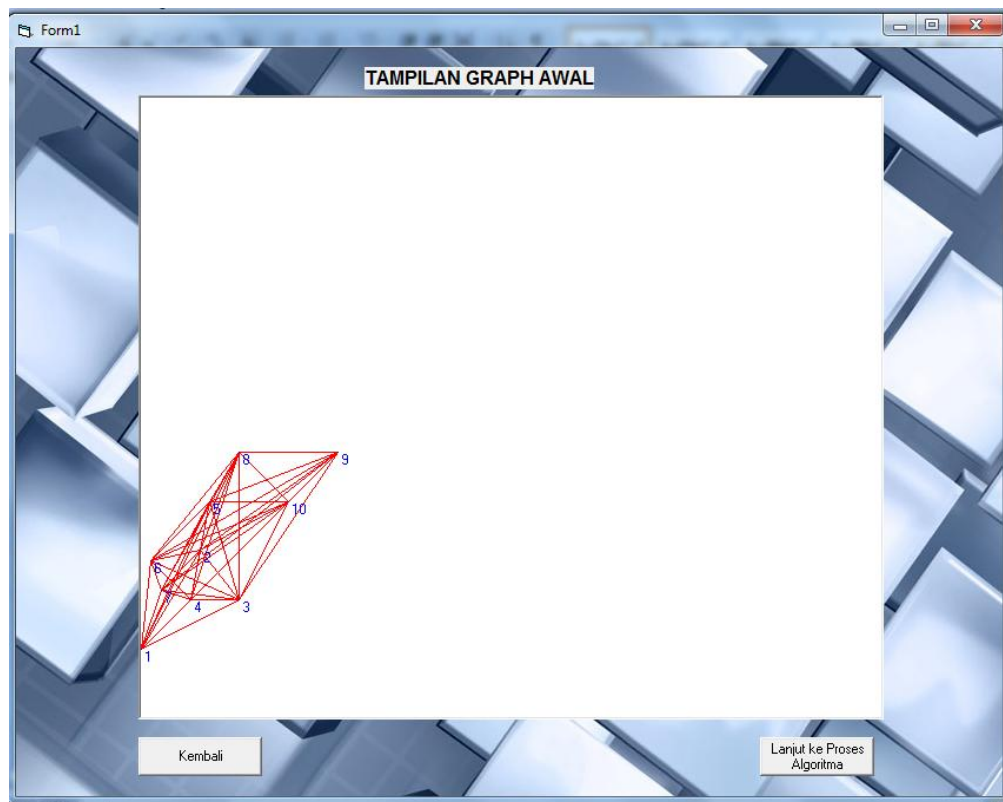
Percobaan	Simpul	Populasi	Crossover	Mutasi	Kelestarian	MST Kruskal	Waktu	MST Genetika	Waktu
18	5	15	0.2	0.1	1	22.27	0.015212	22.27	1.450371
19	5	15	0.2	0.2	1	22.27	0.015972	22.27	1.489012
20	5	15	0.2	0.3	1	22.27	0.014935	22.27	1.563858
21	5	15	0.3	0	1	22.27	0.009088	23.57	1.059099
22	5	15	0.3	0.1	1	22.27	0.017445	22.27	1.49215
23	5	15	0.3	0.2	1	22.27	0.015545	22.27	1.541284
24	5	15	0.3	0.3	1	22.27	0.011326	22.27	1.630786
25	5	25	0.1	0	1	22.27	0.013124	22.27	2.578034
26	5	25	0.1	0.1	1	22.27	0.015398	22.27	2.905628
27	5	25	0.1	0.2	1	22.27	0.015299	22.27	2.8049
28	5	25	0.1	0.3	1	22.27	0.012852	22.27	2.990838
29	5	25	0.2	0	1	22.27	0.015185	22.27	2.640364
30	5	25	0.2	0.1	1	22.27	0.016134	22.27	2.886626
31	5	25	0.2	0.2	1	22.27	0.01384	22.27	2.884284
32	5	25	0.2	0.3	1	22.27	0.013755	22.27	3.079828
33	5	25	0.3	0	1	22.27	0.01479	22.27	2.751301
34	5	25	0.3	0.1	1	22.27	0.01576	22.27	2.908415
35	5	25	0.3	0.2	1	22.27	0.015776	22.27	2.884886
36	5	25	0.3	0.3	1	22.27	0.007896	22.27	2.993097
37	5	50	0.1	0	1	22.27	0.015463	22.27	5.393328
38	5	50	0.1	0.1	1	22.27	0.015098	22.27	4.997313
39	5	50	0.1	0.2	1	22.27	0.016934	22.27	4.094538

Percobaan	Simpul	Populasi	Crossover	Mutasi	Kelestarian	MST Kruskal	Waktu	MST Genetika	Waktu
40	5	50	0.1	0.3	1	22.27	0.009951	22.27	4.458957
41	5	50	0.2	0	1	22.27	0.014371	22.27	4.544754
42	5	50	0.2	0.1	1	22.27	0.015985	22.27	4.295513
43	5	50	0.2	0.2	1	22.27	0.015985	22.27	4.295513
44	5	50	0.2	0.3	1	22.27	0.015985	22.27	4.295513
45	5	50	0.3	0	1	22.27	0.010921	22.27	5.637114
46	5	50	0.3	0.1	1	22.27	0.016648	22.27	4.429647
47	5	50	0.3	0.2	1	22.27	0.011177	22.27	4.957995
48	5	50	0.3	0.3	1	22.27	0.012811	22.27	4.6132

2. Data 10 Simpul

Simpul	X	Y
1	0	0
2	6	10
3	10	5
4	5	5
5	7	15
6	1	9
7	2	6
8	10	20
9	20	20
10	15	15

Graf Awal :



Jumlah Simpul : 10
 Populasi : 2x jumlah simpul s/d 10 x Jumlah Simpul
Crossover : 0.1 - 0.3
 Mutasi : 0 - 0.3
 Kelestarian : 1

Percobaan	Simpul	Populasi	Crossover	Mutasi	Kelestarian	MST Kruskal	Waktu	MST Genetika	Waktu
1	10	20	0.1	0	1	47.81	0.019141	67.59	2.055227
2	10	20	0.1	0.1	1	47.81	0.014851	67.99	2.897161
3	10	20	0.1	0.2	1	47.81	0.014878	67.93	3.068644
4	10	20	0.1	0.3	1	47.81	0.022677	62.5	3.022044
5	10	20	0.2	0	1	47.81	0.011213	66.53	2.337642
6	10	20	0.2	0.1	1	47.81	0.014034	59.65	3.601289
7	10	20	0.2	0.2	1	47.81	0.014499	66.52	3.4633
8	10	20	0.2	0.3	1	47.81	0.020422	65.02	3.564517
9	10	20	0.3	0	1	47.81	0.017452	70.37	2.282563
10	10	20	0.3	0.1	1	47.81	0.013159	58.94	3.43259
11	10	20	0.3	0.2	1	47.81	0.011005	61.1	4.031512
12	10	20	0.3	0.3	1	47.81	0.014351	64.44	4.013601
13	10	30	0.1	0	1	47.81	0.013108	69	1.868174
14	10	30	0.1	0.1	1	47.81	0.025616	61.38	4.671544
15	10	30	0.1	0.2	1	47.81	0.015939	61.56	7.111029
16	10	30	0.1	0.3	1	47.81	0.01635	61.31	6.759636

Percobaan	Simpul	Populasi	Crossover	Mutasi	Kelestarian	MST Kruskal	Waktu	MST Genetika	Waktu
17	10	30	0.2	0	1	47.81	0.016131	67.29	4.977551
18	10	30	0.2	0.1	1	47.81	0.015207	59.03	7.393107
19	10	30	0.2	0.2	1	47.81	0.015943	62.11	6.889143
20	10	30	0.2	0.3	1	47.81	0.016484	66.55	7.793473
21	10	30	0.3	0	1	47.81	0.015343	64.54	4.875199
22	10	30	0.3	0.1	1	47.81	0.016423	62.82	7.270913
23	10	30	0.3	0.2	1	47.81	0.015706	61.39	8.094891
24	10	30	0.3	0.3	1	47.81	0.020581	55.16	9.756081
25	10	50	0.1	0	1	47.81	0.016405	62.38	8.306678
26	10	50	0.1	0.1	1	47.81	0.015649	64.03	11.945444
27	10	50	0.1	0.2	1	47.81	0.021422	62.15	16.927991
28	10	50	0.1	0.3	1	47.81	0.013119	62.23	9.743064
29	10	50	0.2	0	1	47.81	0.016876	60.73	12.845265
30	10	50	0.2	0.1	1	47.81	0.019753	60.88	11.225145
31	10	50	0.2	0.2	1	47.81	0.014301	64.24	11.203996
32	10	50	0.2	0.3	1	47.81	0.022096	58.27	17.412932
33	10	50	0.3	0	1	47.81	0.016584	66.12	7.805667
34	10	50	0.3	0.1	1	47.81	0.017342	58.89	12.714231
35	10	50	0.3	0.2	1	47.81	0.015482	62.16	11.988633
36	10	50	0.3	0.3	1	47.81	0.015853	59.32	11.764714
37	10	100	0.1	0	1	47.81	0.022518	62.38	11.72145
38	10	100	0.1	0.1	1	47.81	0.018731	59.28	18.304056

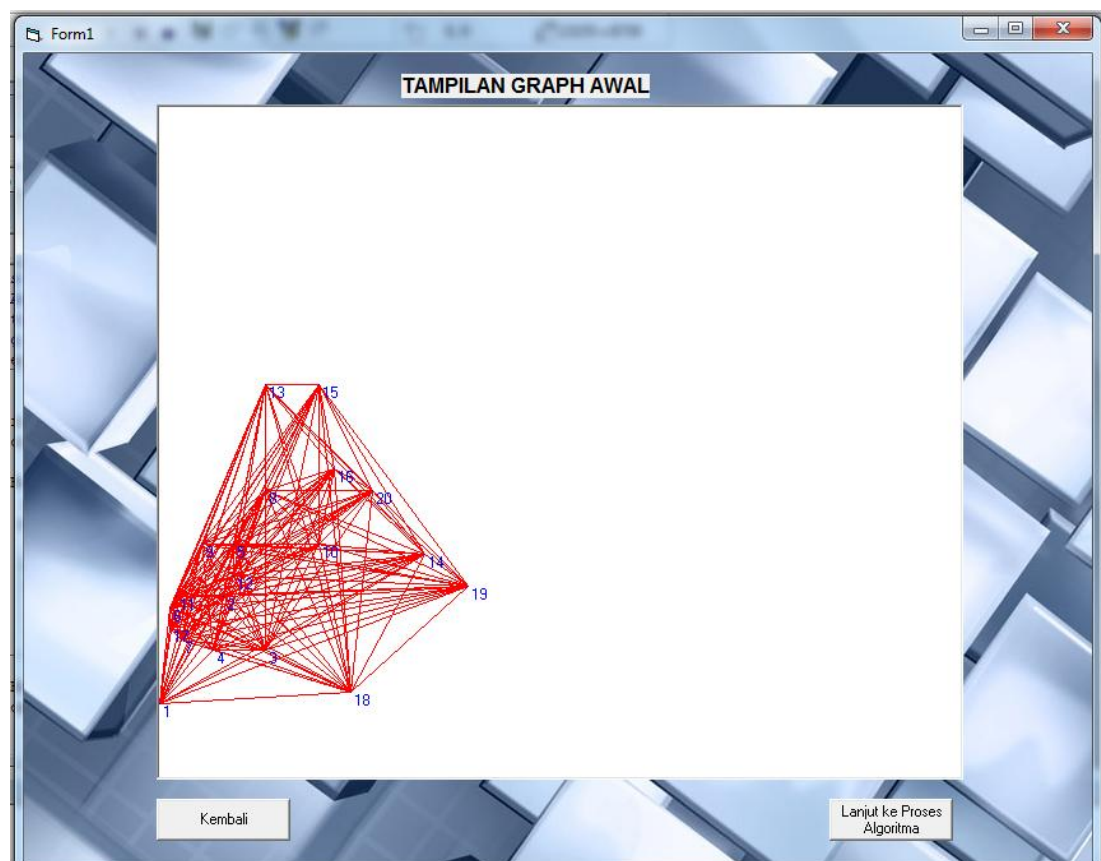
Percobaan	Simpul	Populasi	Crossover	Mutasi	Kelestarian	MST Kruskal	Waktu	MST Genetika	Waktu
39	10	100	0.1	0.2	1	47.81	0.011781	54.31	17.857295
40	10	100	0.1	0.3	1	47.81	0.012026	61.24	20.417556
41	10	100	0.2	0	1	47.81	0.026831	59.19	11.668771
42	10	100	0.2	0.1	1	47.81	0.01469	60.73	19.076944
43	10	100	0.2	0.2	1	47.81	0.014676	60.04	21.097874
44	10	100	0.2	0.3	1	47.81	0.017687	61.35	18.128492
45	10	100	0.3	0	1	47.81	0.031313	55.84	13.048139
46	10	100	0.3	0.1	1	47.81	0.024963	58.59	18.172503
47	10	100	0.3	0.2	1	47.81	0.018952	59.93	18.409427
48	10	100	0.3	0.3	1	47.81	0.019344	58.02	20.56272

3. Data 20 Simpul

Simpul	X	Y
1	0	0
2	6	10
3	10	5
4	5	5
5	7	15
6	1	9
7	2	6
8	10	20
9	4	15
10	15	15

Simpul	X	Y
11	1.5	10
12	7	12
13	10	30
14	25	14
15	15	30
16	16.5	22
17	1	7
18	18	1
19	29	11
20	20	20

Graf awal :



Jumlah Simpul : 20
 Populasi : 2x jumlah simpul s/d 10 x Jumlah Simpul
Crossover : 0.1 - 0.3
 Mutasi : 0 - 0.3
 Kelestarian : 1

Percobaan	Simpul	Populasi	Crossover	Mutasi	Kelestarian	MST Kruskal	Waktu	MST Genetika	Waktu
1	20	40	0.1	0	1	90.37	0.038573	198.01	19.130361
2	20	40	0.1	0.1	1	90.37	0.045922	192.79	56.484864
3	20	40	0.1	0.2	1	90.37	0.031292	195.02	41.304726
4	20	40	0.1	0.3	1	90.37	0.031716	184.96	29.008878
5	20	40	0.2	0	1	90.37	0.030697	192.6	19.066396
6	20	40	0.2	0.1	1	90.37	0.051897	182.11	33.807758
7	20	40	0.2	0.2	1	90.37	0.048622	188.85	35.894702
8	20	40	0.2	0.3	1	90.37	0.051341	172.33	36.954687
9	20	40	0.3	0	1	90.37	0.056464	197.27	22.208159
10	20	40	0.3	0.1	1	90.37	0.028491	182.34	38.0757
11	20	40	0.3	0.2	1	90.37	0.055589	186.78	26.987686
12	20	40	0.3	0.3	1	90.37	0.03854	180.29	25.792134
13	20	60	0.1	0	1	90.37	0.032309	198.87	15.778038
14	20	60	0.1	0.1	1	90.37	0.038124	199.47	42.446791
15	20	60	0.1	0.2	1	90.37	0.03142	181.84	48.070125
16	20	60	0.1	0.3	1	90.37	0.039192	191.3	55.780572
17	20	60	0.2	0	1	90.37	0.039011	179.7	35.412907

Percobaan	Simpul	Populasi	Crossover	Mutasi	Kelestarian	MST Kruskal	Waktu	MST Genetika	Waktu
18	20	60	0.2	0.1	1	90.37	0.032597	178.58	48.357713
19	20	60	0.2	0.2	1	90.37	0.038811	180.42	37.360959
20	20	60	0.2	0.3	1	90.37	0.031172	182.97	61.297579
21	20	60	0.3	0	1	90.37	0.038463	192.47	25.860447
22	20	60	0.3	0.1	1	90.37	0.037996	193.22	25.47372
23	20	60	0.3	0.2	1	90.37	0.038384	191.38	37.407813
24	20	60	0.3	0.3	1	90.37	0.063876	179.37	52.738372
25	20	100	0.1	0	1	90.37	0.038979	174.9	42.083955
26	20	100	0.1	0.1	1	90.37	0.036687	175.75	78.155848
27	20	100	0.1	0.2	1	90.37	0.030464	183.13	78.258003
28	20	100	0.1	0.3	1	90.37	0.046912	179.17	84.082628
29	20	100	0.2	0	1	90.37	0.038545	191.45	63.491682
30	20	100	0.2	0.1	1	90.37	0.032576	177.06	85.888547
31	20	100	0.2	0.2	1	90.37	0.037696	189.44	63.218995
32	20	100	0.2	0.3	1	90.37	0.037785	174.7	64.442224
33	20	100	0.3	0	1	90.37	0.031139	184.73	66.619579
34	20	100	0.3	0.1	1	90.37	0.03184	181.33	77.812252
35	20	100	0.3	0.2	1	90.37	0.031536	182.19	82.276748
36	20	100	0.3	0.3	1	90.37	0.03163	180.09	81.76828
37	20	200	0.1	0	1	90.37	0.031042	179.49	107.720999
38	20	200	0.1	0.1	1	90.37	0.043787	183.37	148.24978
39	20	200	0.1	0.2	1	90.37	0.039964	184.62	152.514745

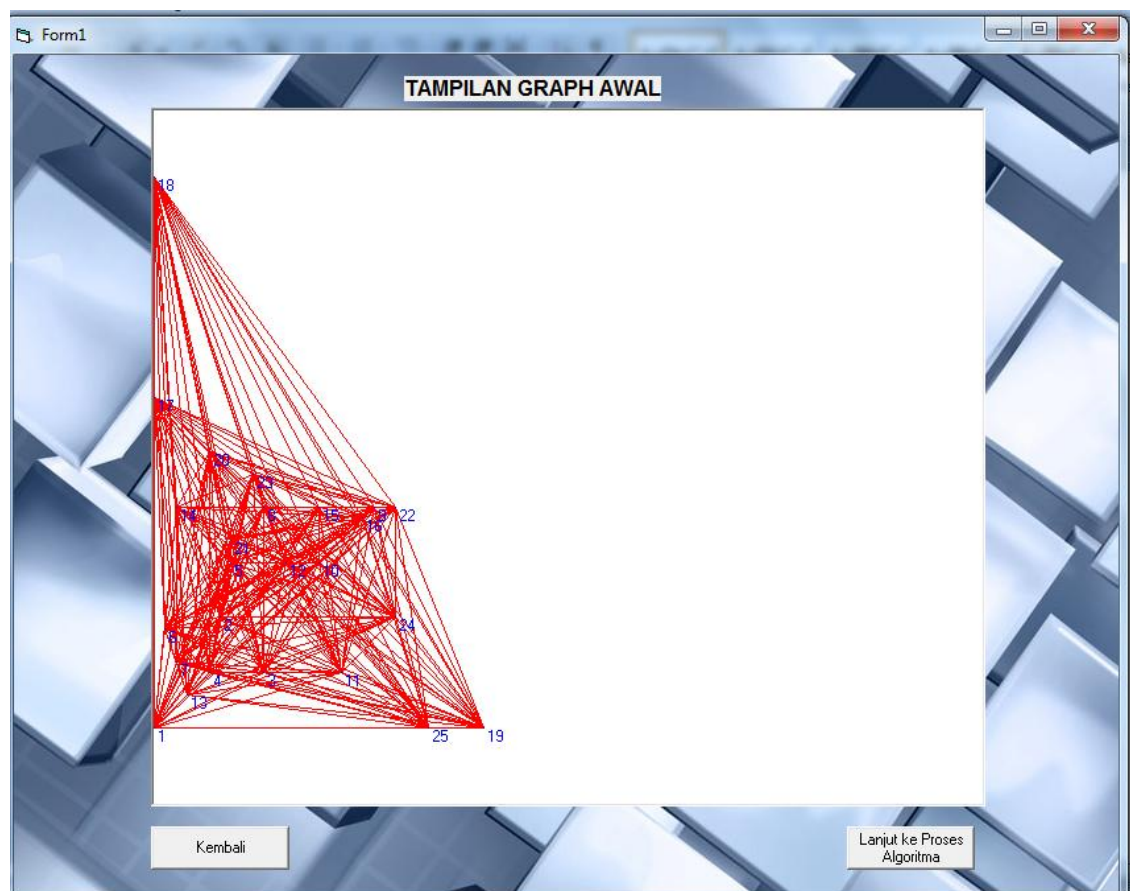
Percobaan	Simpul	Populasi	Crossover	Mutasi	Kelestarian	MST Kruskal	Waktu	MST Genetika	Waktu
40	20	200	0.1	0.3	1	90.37	0.034143	174.68	100.520222
41	20	200	0.2	0	1	90.37	0.037978	180.5	96.509984
42	20	200	0.2	0.1	1	90.37	0.050997	179.02	195.883643
43	20	200	0.2	0.2	1	90.37	0.039108	178.58	165.617664
44	20	200	0.2	0.3	1	90.37	0.040468	187.42	171.180528
45	20	200	0.3	0	1	90.37	0.038755	167.26	100.300446
46	20	200	0.3	0.1	1	90.37	0.038135	173.62	169.028189
47	20	200	0.3	0.2	1	90.37	0.038776	175.97	158.01347
48	20	200	0.3	0.3	1	90.37	0.050511	160.14	163.118443

4. Data 25 Simpul

Simpul	X	Y
1	0	0
2	6	10
3	10	5
4	5	5
5	7	15
6	1	9
7	2	6
8	10	20
9	20	20
10	15	15
11	17	5
12	12	15
13	3	3
14	2	20
15	15	20

Simpul	X	Y
16	19	19
17	0	30
18	0	50
19	30	0
20	5	25
21	7	17
22	22	20
23	9	23
24	22	10
25	25	0

Graf Awal :



Jumlah Simpul : 25
 Populasi : 2x jumlah simpul s/d 10 x Jumlah Simpul
Crossover : 0.1 - 0.3
 Mutasi : 0 - 0.3
 Kelestarian : 1

Percobaan	Simpul	Populasi	Crossover	Mutasi	Kelestarian	MST Kruskal	Waktu	MST Genetika	Waktu
1	25	50	0.1	0	1	124.39	0.072605	278.72	48.369487
2	25	50	0.1	0.1	1	124.39	0.081283	273.82	58.830765
3	25	50	0.1	0.2	1	124.39	0.046995	285.74	37.898213
4	25	50	0.1	0.3	1	124.39	0.056774	298.46	37.811067
5	25	50	0.2	0	1	124.39	0.047889	295.28	24.642417
6	25	50	0.2	0.1	1	124.39	0.056069	280.12	25.5537
7	25	50	0.2	0.2	1	124.39	0.04815	281.51	39.167367
8	25	50	0.2	0.3	1	124.39	0.053932	294.56	78.384746
9	25	50	0.3	0	1	124.39	0.049414	284.65	31.836111
10	25	50	0.3	0.1	1	124.39	0.055147	283.39	66.455537
11	25	50	0.3	0.2	1	124.39	0.045734	288.19	42.129872
12	25	50	0.3	0.3	1	124.39	0.055717	284.36	47.091402
13	25	75	0.1	0	1	124.39	0.048193	304.34	39.989444
14	25	75	0.1	0.1	1	124.39	0.0513	283.89	80.884473
15	25	75	0.1	0.2	1	124.39	0.053055	282.58	77.721702
16	25	75	0.1	0.3	1	124.39	0.048953	293.34	69.343952
17	25	75	0.2	0	1	124.39	0.047643	296.18	38.085072

Percobaan	Simpul	Populasi	Crossover	Mutasi	Kelestarian	MST Kruskal	Waktu	MST Genetika	Waktu
18	25	75	0.2	0.1	1	124.39	0.048247	296.04	77.612856
19	25	75	0.2	0.2	1	124.39	0.068342	274.82	69.727658
20	25	75	0.2	0.3	1	124.39	0.047186	284.05	61.660476
21	25	75	0.3	0	1	124.39	0.047038	292.35	34.822519
22	25	75	0.3	0.1	1	124.39	0.046206	287.83	73.484067
23	25	75	0.3	0.2	1	124.39	0.044964	283.89	61.127379
24	25	75	0.3	0.3	1	124.39	0.05249	288.59	58.318197
25	25	125	0.1	0	1	124.39	0.051536	287.1	46.456867
26	25	125	0.1	0.1	1	124.39	0.051212	272.38	77.256242
27	25	125	0.1	0.2	1	124.39	0.048168	302.67	79.256123
28	25	125	0.1	0.3	1	124.39	0.05156	293.2	68.973181
29	25	125	0.2	0	1	124.39	0.045356	274.54	46.945004
30	25	125	0.2	0.1	1	124.39	0.049609	273.57	70.342079
31	25	125	0.2	0.2	1	124.39	0.046376	284.39	88.285153
32	25	125	0.2	0.3	1	124.39	0.045187	273.85	73.653784
33	25	125	0.3	0	1	124.39	0.047159	280.88	54.426632
34	25	125	0.3	0.1	1	124.39	0.049765	271.5	76.238793
35	25	125	0.3	0.2	1	124.39	0.048749	276.96	91.716399
36	25	125	0.3	0.3	1	124.39	0.045461	310.23	86.355647
37	25	250	0.1	0	1	124.39	0.045494	288.91	170.775998
38	25	250	0.1	0.1	1	124.39	0.050822	283.34	232.755311
39	25	250	0.1	0.2	1	124.39	0.045903	275.43	226.198832

Percobaan	Simpul	Populasi	Crossover	Mutasi	Kelestarian	MST Kruskal	Waktu	MST Genetika	Waktu
40	25	250	0.1	0.3	1	124.39	0.048953	291.41	276.6628
41	25	250	0.2	0	1	124.39	0.04779	267.58	200.525442
42	25	250	0.2	0.1	1	124.39	0.048044	287.67	292.697868
43	25	250	0.2	0.2	1	124.39	0.048611	277.1	261.844396
44	25	250	0.2	0.3	1	124.39	0.047671	275.77	244.617075
45	25	250	0.3	0	1	124.39	0.048325	274.84	161.318054
46	25	250	0.3	0.1	1	124.39	0.049003	259.91	284.00552
47	25	250	0.3	0.2	1	124.39	0.046552	275.19	231.806031
48	25	250	0.3	0.3	1	124.39	0.049938	279.88	242.042099